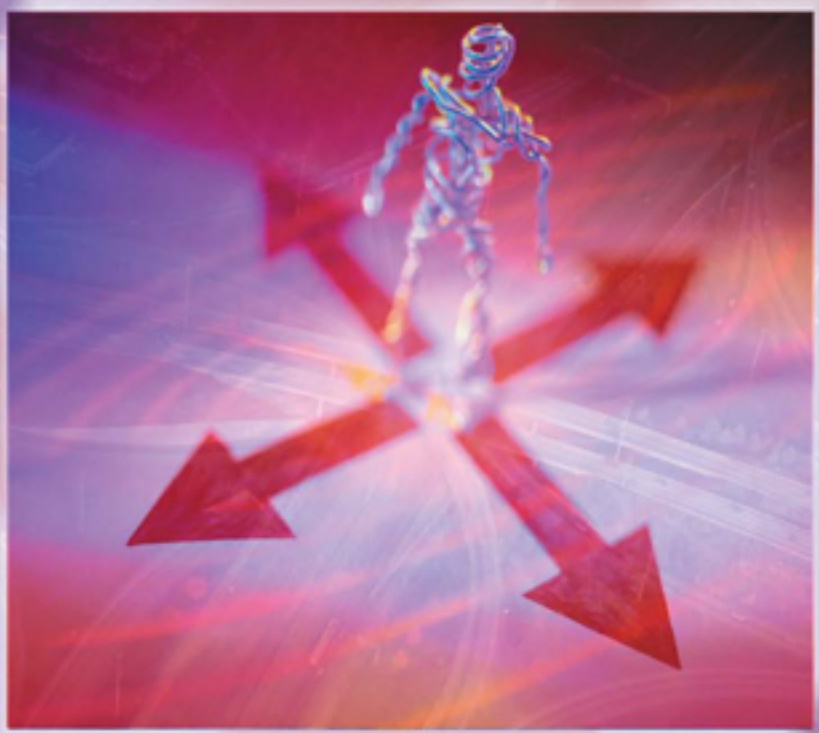
VISIT…

AUERBACH

TCP/IP Professional Reference Guide



GILBERT HELD

TCP/IP Professional Reference Guide

OTHER AUERBACH PUBLICATIONS

A Standard for Auditing Computer Applications
Martin Krist
ISBN: 0-8493-9983-1

A Technical Guide to IPSec Virtual Private Networks
James S. Tiller
ISBN: 0-8493-0876-3

Analyzing Business Information Systems
Shouhong Wang
ISBN: 0-8493-9240-3

Broadband Networking
James Trulove, Editor
ISBN: 0-8493-9821-5

Communications Systems Management Handbook, 6th Edition
Anura Gurugé and
Lisa M. Lindgren, Editors
ISBN: 0-8493-9826-6

Computer Telephony Integration
William Yarberry, Jr.
ISBN: 0-8493-9995-5

Data Management Handbook 3rd Edition
Sanjiv Purba, Editor
ISBN: 0-8493-9832-0

Electronic Messaging
Nancy Cox, Editor
ISBN: 0-8493-9825-8

Enterprise Operations Management Handbook, 2nd Edition
Steve F. Blanding, Editor
ISBN: 0-8493-9824-X

Enterprise Systems Architectures
Andersen Consulting
ISBN: 0-8493-9836-3

Enterprise Systems Integration
John Wyzalek, Editor
ISBN: 0-8493-9837-1

Healthcare Information Systems
Phillip L. Davidson, Editor
ISBN: 0-8493-9963-7

Information Security Architecture
Jan Tudor Killmeyer
ISBN: 0-8493-9988-2

Information Security Management Handbook, 4th Edition, Volume 2
Harold F. Tipton and Micki Krause, Editors
ISBN: 0-8493-0800-3

IS Management Handbook, 7th Edition
Carol V. Brown, Editor
ISBN: 0-8493-9820-7

Information Technology Control and Audit
Frederick Gallegos, Sandra Allen-Senft, and Daniel P. Manson
ISBN: 0-8493-9994-7

Internet Management
Jessica Keyes, Editor
ISBN: 0-8493-9987-4

Local Area Network Handbook, 6th Edition
John P. Slone, Editor
ISBN: 0-8493-9838-X

Multi-Operating System Networking: Living with UNIX, NetWare, and NT
Raj Rajagopal, Editor
ISBN: 0-8493-9831-2

TCP/IP Professional Reference Guide
Gilbert Held
ISBN: 0-8493-0824-0

The Network Manager's Handbook, 3rd Edition
John Lusa, Editor
ISBN: 0-8493-9841-X

Project Management
Paul C. Tinnirello, Editor
ISBN: 0-8493-9998-X

Effective Use of Teams in IT Audits,
Martin Krist
ISBN: 0-8493-9828-2

Systems Development Handbook, 4th Edition
Paul C. Tinnirello, Editor
ISBN: 0-8493-9822-3

AUERBACH PUBLICATIONS

www.auerbach-publications.com
TO Order: Call: 1-800-272-7737 • Fax: 1-800-374-3401
E-mail: orders@crcpress.com

TCP/IP Professional Reference Guide

GILBERT HELD

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Held, Gilbert, 1943-
TCP/IP professional reference guide / Gilbert Held.
p. cm.
Includes bibliographical references and index.
ISBN 0-8493-0824-0 (alk. paper)
1. TCP/IP (Computer network protocol) 2. Internetworking (Telecommunication)
I. Title.

TK5105.585 .H448 2000
004.6′2—dc21 00-045511

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

© 2001 by CRC Press LLC
Auerbach is an imprint of CRC Press LLC

No claim to original U.S. Government works
International Standard Book Number 0-8493-0824-0
Library of Congress Card Number 00-045511
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Contents

Chapter 1 Overview 1
Applications 1
Current Applications 2
Electronic Mail 2
File Transfers 5
Remote Terminal Access 5
Web Surfing 8
Emerging Applications 8
Audio and Video Players 8
Voice Over IP 10
Virtual Private Networking 10
Book Preview 12
The Protocol Suite 12
The Standards Process 12
The Internet Protocol and Related Protocols 13
Transport Layer Protocols 13
Applications and Built-in Diagnostic Tools 13
Routing 13
Security 13
Emerging Technologies 14

Chapter 2 The Protocol Suite 15
The ISO Reference Model 15
OSI Reference Model Layers 16
Layer 1: The Physical Layer 16
Layer 2: The Data Link Layer 17
Layer 3: The Network Layer 17
Layer 4: The Transport Layer 18
Layer 5: The Session Layer 18
Layer 6: The Presentation Layer 19
Layer 7: The Application Layer 19
Data Flow 19

The TCP/IP Protocol Suite 19
The Network Layer 20
IP 20
ARP 21
ICMP 21
The Transport Layer 21
TCP 21
UDP 22
Application Layer 23
Data Flow 23

Chapter 3 Internet Governing Bodies and the Standards Process 25
Internet Governing Bodies 25
Internet Evolution 25
The IAB and IETF 27
The IANA 27
Request for Comments 28
The Standards Process 28
Draft RFC 28
Proposed Standard and Draft Standard 28
RFC Standard 29
RFC Details 29
RFC Categories 29
Accessing RFCs 29

Chapter 4 The Internet Protocol and Related Protocols 37
The Internet Protocol 38
Datagrams and Segments 38
Datagrams and Datagram Transmission 38
Routing 39
The IP Header 39
Bytes Versus Octets 39
Vers Field 40
Hlen Field 40
Service Type Field 41
Total Length Field 42
Identification and Fragment Offset Fields 43
Flag Field 44
Time to Live Field 44
Protocol Field 44
Header Checksum Field 45
Source and Destination Address Fields 45
IP Addressing 48
Overview 49
The IP Addressing Scheme 50
Address Changes 50
Rationale 51
Overview 52

Class A Addresses 53
Loopback 53
Class B Addresses 54
Class C Addresses 56
Class D Addresses 56
Class E Addresses 57
Dotted Decimal Notation 58
Basic Workstation Configuration 58
Reserved Addresses 62
Subnetting 64
Overview 64
Subnetting Example 64
Host Restrictions 66
The Zero Subnet 66
Internal Versus External Subnet Viewing 67
Using the Subnet Mask 68
Multiple Interface Addresses 71
Address Resolution 72
Ethernet and Token Ring Frame Formats 72
LAN Delivery 73
Address Resolution Operation 73
ARP Packet Fields 74
Locating the Required Address 74
Gratuitous ARP 75
Proxy ARP 75
RARP 75
ICMP 76
Overview 76
The ICMP Type Field 76
The ICMP Code Field 78
Evolution 78

Chapter 5 The Transport Layer 81
TCP 81
The TCP Header 81
Source and Destination Port Fields 82
Multiplexing and Demultiplexing 83
Port Numbers 84
Well-Known Ports 84
Registered Ports 84
Dynamic or Private Ports 84
Sequence and Acknowledgment Number Fields 84
Hlen Field 86
Code Bits Field 87
URG Bit 87
ACK Bit 87
PSH Bit 87
RST Bit 87
SYN Bit 88
FIN Bit 88

Window Field 88
Checksum Field 88
Urgent Pointer Field 88
Options 89
Padding Field 89
Connection Establishment 89
Connection Function Calls 89
Port Hiding 90
Passive OPEN 90
Active OPEN 90
The Three-Way Handshake 91
Overview 91
Operation 91
The TCP Window 93
Avoiding Congestion 94
TCP Slow Start 94
The Slow Start Threshold 95
TCP Retransmissions 96
Session Termination 96
UDP 96
The UDP Header 97
Source and Destination Port Fields 98
Length Field 98
Checksum Field 98
Operation 98
Applications 99

Chapter 6 Applications and Built-in Diagnostic Tools 101
The DNS 101
Purpose 101
The Domain Name Structure 102
The Domain Name Tree 102
The Name Resolution Process 103
Data Flow 104
Time Consideration 105
DNS Records 105
The SOA Record 106
Checking Records 107
Diagnostic Tools 107
Ping 107
Operation 107
Implementation 108
Using Windows NT Ping 109
Resolution Time Considerations 110
Applications 110
Traceroute 111
Operation 111
Using Microsoft Windows Tracert 112
Tracing a Route 113
Applications 114

NSLOOKUP ... 114
Operation ... 114
Finding Information about Mail Servers at Yale ... 116
Viewing the SOA Record ... 116
Protecting Server Information ... 117
Finger ... 118
Format ... 118
Security Considerations ... 118
Applications ... 119

Chapter 7 Routing and Routing Protocols ... 121
Network Routing ... 122
Routing in a Global System ... 122
Autonomous Systems ... 122
Types of Routing Protocols ... 124
Need for Routing Tables ... 125
Routing Table Update Methods ... 127
The Routing Information Protocol ... 128
Illustrative Network ... 128
Dynamic Table Updates ... 128
Basic Limitations ... 131
RIP Versions ... 131
The Basic RIPv1 Packet ... 132
Command Field ... 132
Version Field ... 132
Family of Net X Field ... 133
Net X Address Field ... 133
Distance to Network X Field ... 133
RIPv1 Limitations ... 133
RIPv2 ... 133
Route Tag Field ... 134
Next Hop Field ... 134
Authentication Support ... 135
OSPF ... 135
Overview ... 136
Path Metrics ... 136
Initialization Activity ... 136
Router Types ... 137
Message Types ... 137
Type 1 Message ... 138
Type 2 Message ... 138
Type 3 Message ... 138
Type 4 Message ... 138
Type 5 Message ... 139
Type 6 Message ... 139
Operation ... 139

Chapter 8 Security ... 141
Router Access Considerations ... 142
Router Control ... 142

Direct Cabling ... 142
Benefits and Limitations ... 142
Telnet and Web Access ... 143
Protection Limitation ... 143
Router Access Lists ... 146
Rationale for Use ... 146
Ports Govern Data Flow ... 147
Data Flow Direction ... 148
Types of Access Lists ... 148
Standard Access Lists ... 148
Extended Access Lists ... 150
New Capabilities in Access Lists ... 152
Named Access Lists ... 152
Reflexive Access Lists ... 153
Time-based Access Lists ... 155
TCP Intercept ... 156
Applying a Named Access List ... 157
Configuration Principles ... 158
Limitations ... 158
Firewalls ... 158
Installation Location ... 158
Basic Functions ... 159
Proxy Services ... 159
Authentication ... 161
Encryption ... 162
Network Address Translation ... 162

Chapter 9 Emerging Technologies ... 163
Virtual Private Networking ... 163
Benefits ... 163
Reducing Hardware Requirements ... 163
Reliability ... 165
Economics ... 165
Limitations ... 166
Authentication ... 166
Encryption ... 167
Other Issues to Consider ... 167
Setting up Remote Access Service ... 168
Mobile IP ... 171
Overview ... 171
Operation ... 172
Voice over IP ... 173
Constraints ... 174
Latency ... 174
Packet Network Operation ... 175
Voice Digitization Method ... 176
Packet Subdivision ... 177
Networking Configurations ... 177
Router Voice Module Utilization ... 177
Voice Gateway ... 178

IPv6 179
Overview 180
Address Architecture 180
Address Types 180
Address Notation 180
Address Allocation 181
Provider-Based Addresses 181
Special Addresses 182

Appendixes: TCP/IP Protocol Reference Numbers 185
Appendix A: ICMP Type and Code Values 189
Appendix B: Internet Protocol (IP) Protocol Type Field Values 193
Appendix C: Port Numbers 197

Index 231

Preface

The TCP/IP protocol suite has evolved from an academic networking tool to the driving force behind the Internet, intranets, and extranets. Advances in networking and communications hardware based upon the TCP/IP protocol suite are opening a new range of technologies that provides the potential to considerably affect our lives. Such technologies as the new version of the Internet Protocol (IP), referred to as IPv6, the use of virtual private networks (VPNs), the convergence of voice and data through a technology referred to as Voice over IP (VoIP), and the expansion of data transmission over wireless communications (mobile IP) can be expected to govern the manner by which we perform many daily activities. Thus, the TCP/IP protocol suite is dynamically changing to reflect advances in technology, and, to paraphrase an often-used term, can be considered to represent "the protocol for the new millennium."

This book was written as a comprehensive reference to the TCP/IP protocol suite for professionals that need to know a range of protocol-related information. Commencing with an overview of the protocol suite, this book examines the key components of the TCP/IP protocol suite. This examination includes the manner by which the various protocols operate, how applications operate, addressing issues, security methods, routing, and an overview of emerging technologies.

The goal of this book is to explain both the "how" and the "why" of the TCP/IP protocol suite. The "how" refers to how various network protocols and applications operate, as this information can be important for selecting one application over another, as well as for attempting to resolve problems and network capacity issues. The "why" refers to this author's best guess as to the rationale for the structure of the TCP/IP protocol suite and the manner by which various components interact. Although no reader was probably present when various meetings occurred that defined the structure of the TCP/IP protocol suite, a review of the manner by which different components of the suite operate allows one to note why it might have been designed.

This in turn provides a considerable amount of information concerning both how and when to use certain members of the protocol suite.

As a professional author, I highly value reader feedback. Please feel free to contact me through the publisher of this book or e-mail me at gil_held@yahoo.com. Let me know if there are certain topics that you would like to see additional information coverage on, if I omitted a topic of interest, or if I should expand coverage of an existing topic.

Gilbert Held
Macon, Georgia

Acknowledgments

The creation of a book is a team effort that requires the contribution of many people. Thus, I would be remiss if I did not acknowledge the efforts of the many people who were instrumental in converting the writings of this author into the book you are now reading.

It is always important to have the support of the acquisitions editor and publisher; however, it is even better to have a most enthusiastic backing for a writing project. Thus, I would like to thank Theron Shreve and Rich O'Hanley for their enthusiastic endorsement of this book.

As a frequent traveler to the four corners of the world, I often encounter electrical outlets that never quite mate with the various adapter kits that I've purchased. Due to this, my writing productivity is considerably enhanced by using pen and paper, especially at locations where it was only possible to shave with a razor, and a laptop battery had long ago reached an undesirable level of power. While I try to write legibly, this is not always the case. Thus, I am once more indebted to Mrs. Linda Hayes for turning my handwritten notes and sketches into a professional manuscript.

Last, but not least, the preparation of a book is a time-consuming task, requiring many hours of effort on weekends and evenings. Once again I am indebted to my wife, Beverly, for her patience and understanding.

Chapter 1

Overview

The TCP/IP protocol suite has evolved from primarily an academic and research communications protocol into a protocol that affects the lives of most individuals. Although most, if not all, readers are familiar with the Internet, that mother of all networks which represents only one use of the TCP/IP protocol suite. Today, many organizations are creating private networks based on the use of the TCP/IP protocol suite that are referred to as intranets. In addition, the Internet is being used to interconnect geographically separated networks through a technology referred to as Virtual Private Networks (VPNs). Recognizing the versatility of the TCP/IP protocol suite, IP is now being used to transport voice, and the transmission of data over wireless communications is evolving to provide mobile users with the ability to access e-mail and surf the Web from their mobile phones. Thus, the TCP/IP protocol suite can be considered to represent the protocol for the new millennium.

This introductory chapter focuses on the role of the TCP/IP protocol suite. In doing so, the chapter concentrates on common and emerging applications supported by this technology, and takes the reader on a brief tour of the focus of succeeding chapters by previewing those chapters. This information, either by itself or in conjunction with the index, can be used to rapidly locate particular information of interest.

Applications

When the TCP/IP protocol suite was initially developed, it was used to support a relatively small handful of applications. Those applications included electronic mail, file transfer, and remote terminal operations. Since the initial development of the TCP/IP protocol suite, its modular architecture has enabled literally hundreds of applications to be developed that use the protocol suite as a transport for communications. This section briefly reviews a core set of

current and emerging applications to obtain an appreciation for the role of the TCP/IP protocol suite.

TCP/IP applications can be subdivided into three general categories: obsolete or little used, current, and emerging. Although obsolete or little-used applications are interesting from a historical perspective, their value for the networking professional is minimal; thus, for the most part, this book focuses on current and evolving applications.

Current Applications

There is a core set of TCP/IP applications that are used by most persons. Those applications include electronic mail, file transfer, remote terminal operations, and Web surfing. Although not directly used by most people, the domain name service (DNS) is crucial for the operation of TCP/IP-based networks as it provides the translation process between host names and IP addresses. Because the vast majority of people who use TCP/IP-based networks enter host addresses while routing is based on the use of IP addresses, DNS provides the crucial link between the two. The remainder of this section briefly reviews the operation and utilization of the core set of current applications commonly used by people on TCP/IP-based networks. This information is presented to ensure that readers with different networking backgrounds obtain a common level of appreciation for the majority of current applications used on TCP/IP-based networks.

Electronic Mail

The TCP/IP protocol suite dates to the 1960s when government laboratories and research universities required a method to share ideas in an expedient manner. Among the first applications developed for the protocol suite was a text-based electronic mail system.

Over the past 30+ years, the use of electronic mail has evolved from a text-based messaging system into the development of sophisticated, integrated calendar, messaging, and documenting systems that perform electronic mail.

One example of a popular integrated e-mail system is Microsoft's Outlook, whose main screen is illustrated in Exhibit 1.1. Through the use of Outlook, one can send and receive conventional text-based messages, attach graphic images and word processing documents within that message, develop the equivalent of an electronic "Rolodex" via the use of a contact folder, and use its calendar facility as a reminder to perform different tasks.

Exhibit 1.2 illustrates a portion of the additional capability obtained through the use of Outlook. This example uses the program's calendar feature, which enables one to both schedule events as well as define tasks and indicate the status of different tasks. Exhibit 1.2 reveals that a meeting and a videoconference have been scheduled and indicates a task for completion on the indicated day.

Unlike the early versions of electronic mail that depended on the TCP/IP protocol suite for communications, Microsoft's Outlook, as well as such

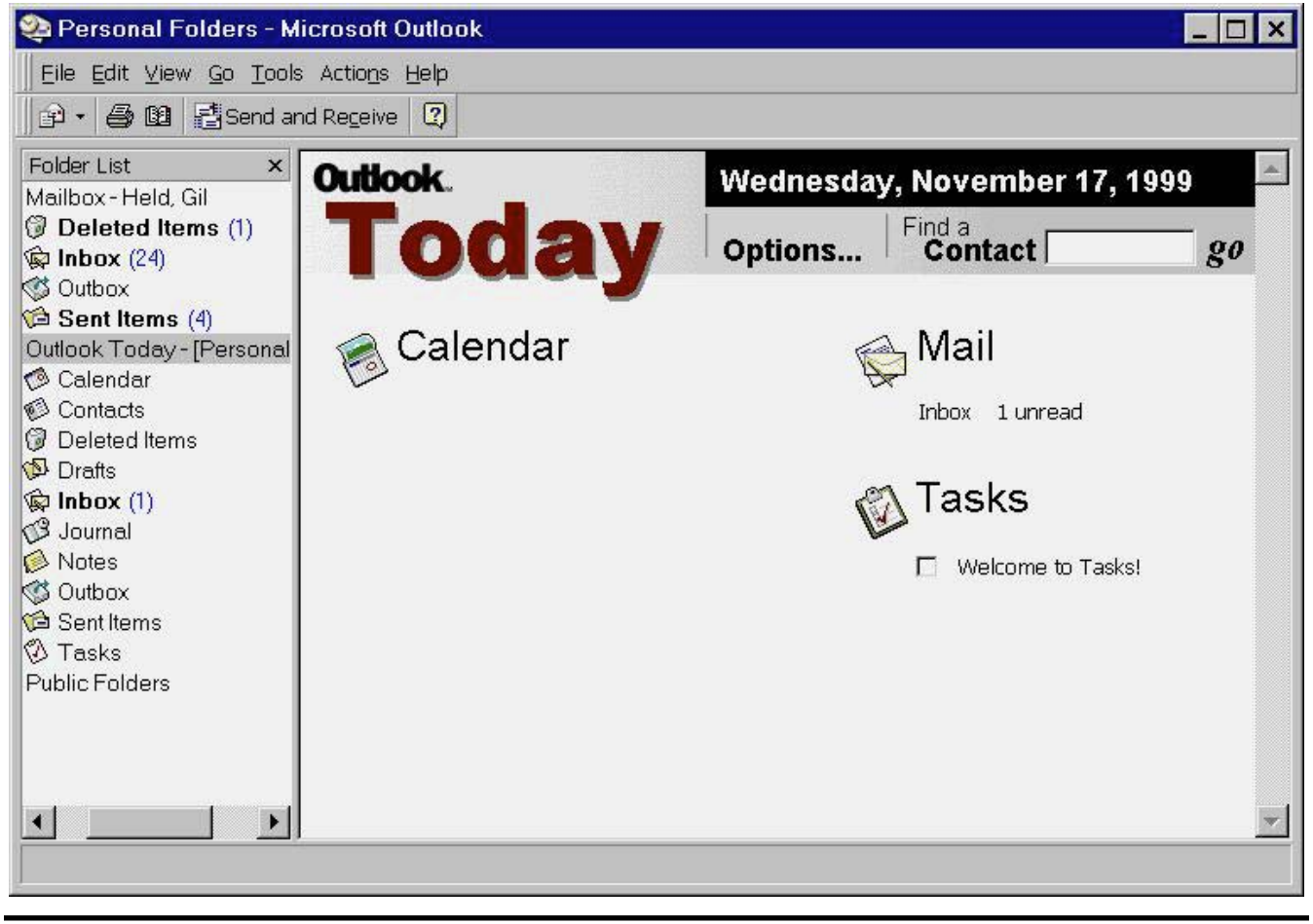


Exhibit 1.1 Viewing the Main Display of Microsoft's Outlook Program

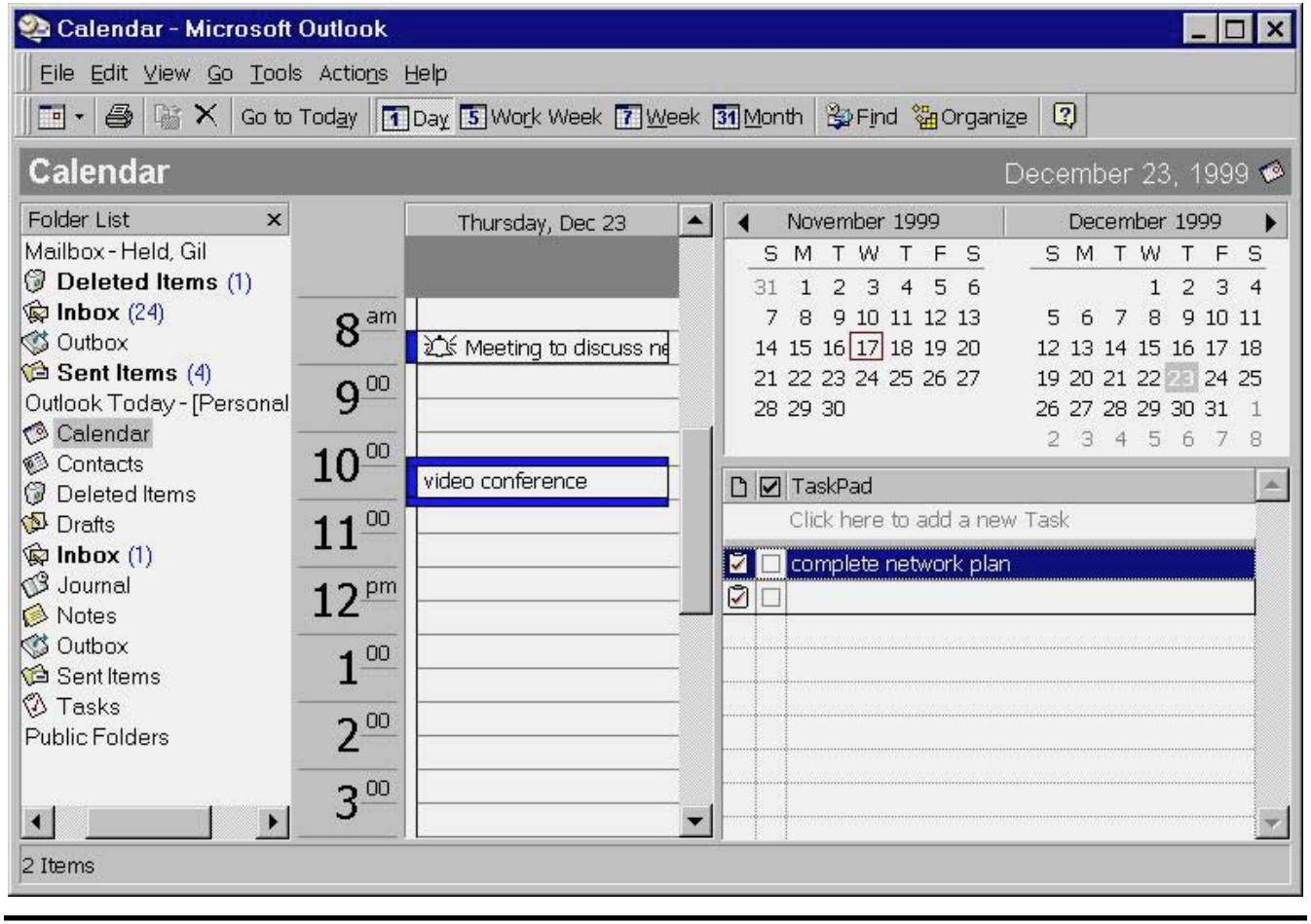


Exhibit 1.2 Using Microsoft's Outlook Calendar and Task Pad

competitive products as Lotus Notes and Novell's GroupWise, support many communications protocols. In fact, just a few years ago, IBM's System Network Architecture (SNA) and Novell's NetWare IPX and SPX protocols accounted for approximately 70 percent of the communications market. The growth in the use of the Internet and the development of corporate intranets has reversed protocol utilization, with the TCP/IP protocol stack now accounting for approximately 70 percent of the communication's market.

File Transfers

A second application that traces its roots to the initial development of the TCP/IP protocol suite is file transfer. During the 1960s, many research laboratories and universities required a mechanism to share large quantities of data, resulting in the development of the File Transfer Protocol (FTP), which more accurately represents an application that facilitates file transfers.

Early versions of FTP applications were text based. Although several software developers introduced graphic user interface versions of FTP during the mid-1990s, the popular Windows operating system added a text-based FTP that represents one of the more popular methods for transferring files.

An example of the use of a Windows FTP application is illustrated in Exhibit 1.3. Note that, with the exception of Windows Version 3.1, all later versions of the ubiquitous Microsoft operating system include FTP as an MS-DOS application. Because it is free, the addition of a TCP/IP protocol stack with the introduction of Windows 95 to include several basic applications caused many third-party software developers that concentrated on TCP/IP applications to undergo a severe contraction in sales. In fact, although there are several graphic user interface versions of FTP available for use, most such products are now shareware instead of commercial products. Thus, the inclusion of the TCP/IP protocol suite in different versions of Windows had a significant impact on the market for stand-alone applications.

Remote Terminal Access

A third core application that dates to the 1960s is remote terminal access in the form of the Telnet application. During the 1960s, it was recognized that a mechanism to access distant computers as if a person's local computer was directly connected to the distant computer would be very desirable. This capability would allow people to configure remote devices as if they were directly connected to the remote device and resulted in the development of the Telnet application.

Exhibit 1.4 illustrates the use of a Telnet application program built into Microsoft's Windows. In this example, Telnet is being used to access a remote router and display information concerning the router's interfaces. Here, the ability to use Telnet saves a trip to the remote router and what would otherwise be a necessity to directly cable a terminal or PC into the router's console port.

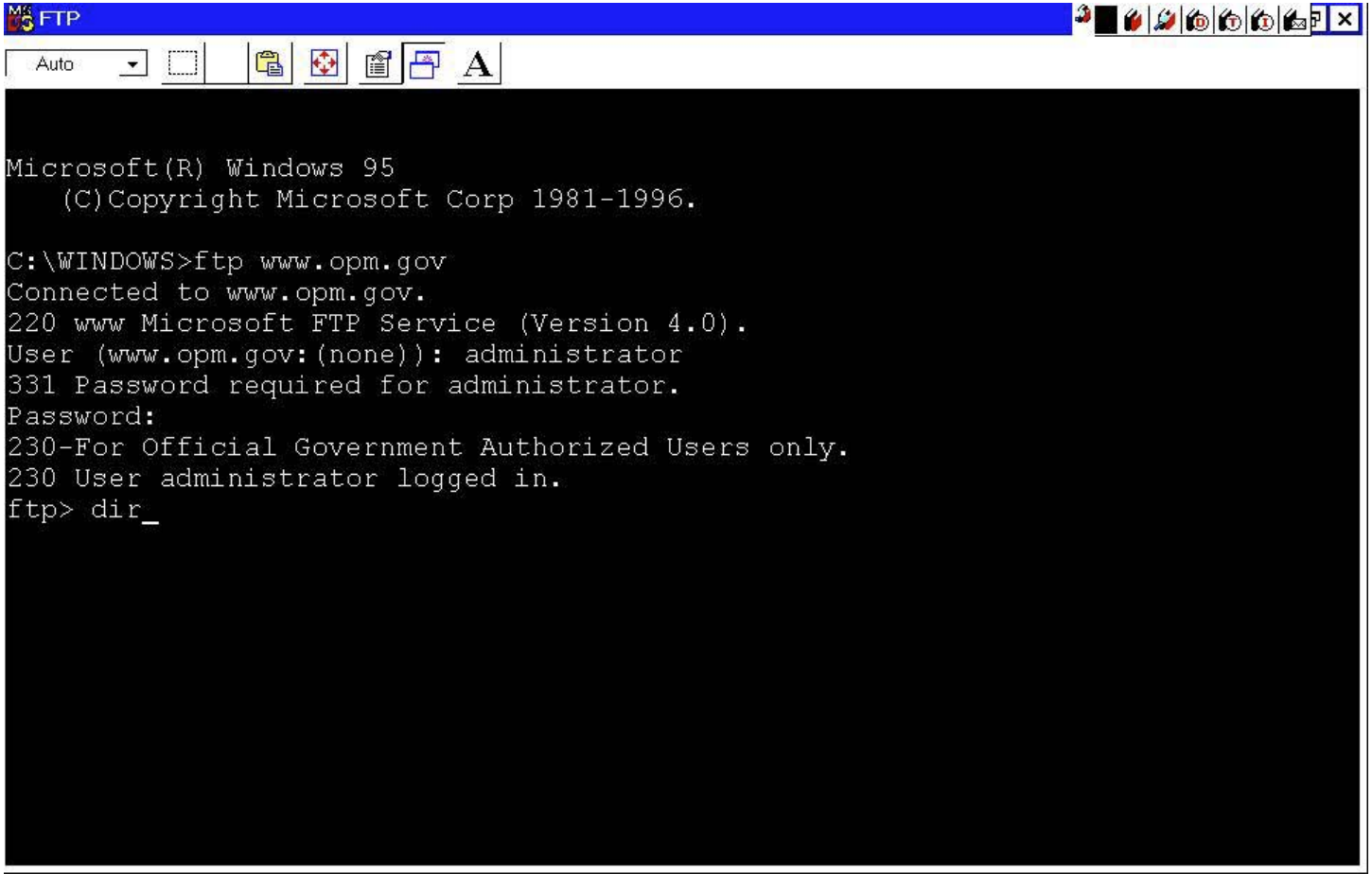


Exhibit 1.3 An Example of the Use of a Windows FTP Application

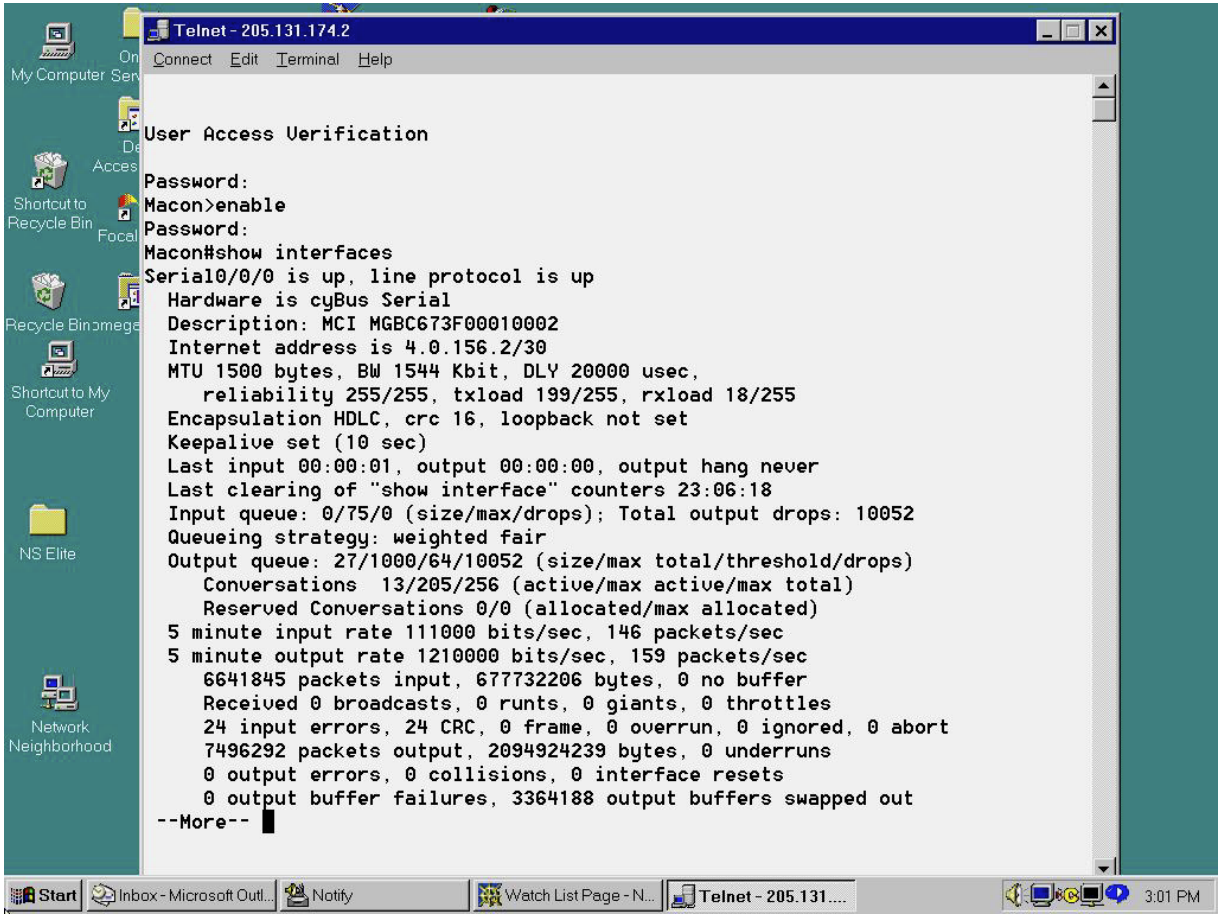

Telnet - 205.131.174.2
Connect Edit Terminal Help
User Access Verification
Password:
Macon>enable
Password:
Macon#show interfaces
Serial0/0/0 is up, line protocol is up
Hardware is cyBus Serial
Description: MCI MGBC673F00010002
Internet address is 4.0.156.2/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 199/255, rxload 18/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive set (10 sec)
Last input 00:00:01, output 00:00:00, output hang never
Last clearing of "show interface" counters 23:06:18
Input queue: 0/75/0 (size/max/drops); Total output drops: 10052
Queueing strategy: weighted fair
Output queue: 27/1000/64/10052 (size/max total/threshold/drops)
Conversations 13/205/256 (active/max active/max total)
Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 111000 bits/sec, 146 packets/sec
5 minute output rate 1210000 bits/sec, 159 packets/sec
6641845 packets input, 677732206 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
24 input errors, 24 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
7496292 packets output, 2094924239 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 3364188 output buffers swapped out
--More--
My Computer
Shortcut to Recycle Bin
Recycle Bin
Shortcut to My Computer
NS Elite
Network Neighborhood
Start
Inbox - Microsoft Outl...
Notify
Watch List Page - N...
Telnet - 205.131....
3:01 PM

Exhibit 1.4 Using Telnet to Access a Remote Router and Determine the State of Its Interfaces

Web Surfing

While it is true that most people correctly associate the use of the Internet with a Web browser, this is only one part of a complex story. The first commercial browser had a limited capability and was primarily used for navigation to different Web sites and the display of Web pages. As Web sites proliferated, they began to add new applications that required browser developers and third-party software developers to add plug-ins to extend the capabilities of browser software. Examples of some common plug-ins include video- and audioconferencing, music playing, and authentication and encryption.

Exhibit 1.5 illustrates the display of the Netscape Communicator menu bar. On examining the entries in the drop-down menu, one notes that this browser more accurately represents an integrated application. Included in the software is a Web browser (Navigator), Web page creation (Composer), and calendar (Calendar) capability, as well as functions for performing conferencing. Looking at the background of the illustration shown in Exhibit 1.5, one notes the display of the home page for Amazon.com, a very popular electronic commerce site that expanded its initial focus from providing consumers with discounts for books to the online retailing of CDs, videos, toys, electronics, and other products. Within just a few years, electronic commerce on the Web grew from under $100 million to over $12 billion, with the TCP/IP protocol suite facilitating the growth in online sales due to the flexibility of the protocol suite to accommodate the new protocols and applications necessary to support electronic commerce.

With an appreciation for the role of a core set of TCP/IP applications, one can now focus on several emerging applications.

Emerging Applications

There are several emerging applications that have the potential to alter the manner by which people perform daily activities. While such applications are interesting from the perspective of a book on the evolution of the TCP/IP protocol suite, one also needs to be aware of emerging applications as they create new demands on network resources. Three emerging TCP/IP applications that deserve mention are audio and video players, the transmission of Voice over IP networks, and the use of virtual private networks.

Audio and Video Players

One of the major benefits of the Internet is its ability to function as a vast distribution center for information. While Web surfing has been very popular for several years, within the past year the distribution of music and the use of audio and video players to provide end users with the ability to convert their PCs into miniature televisions have gained in popularity. One popular example of an emerging application is the RealPlayer from RealNetworks.

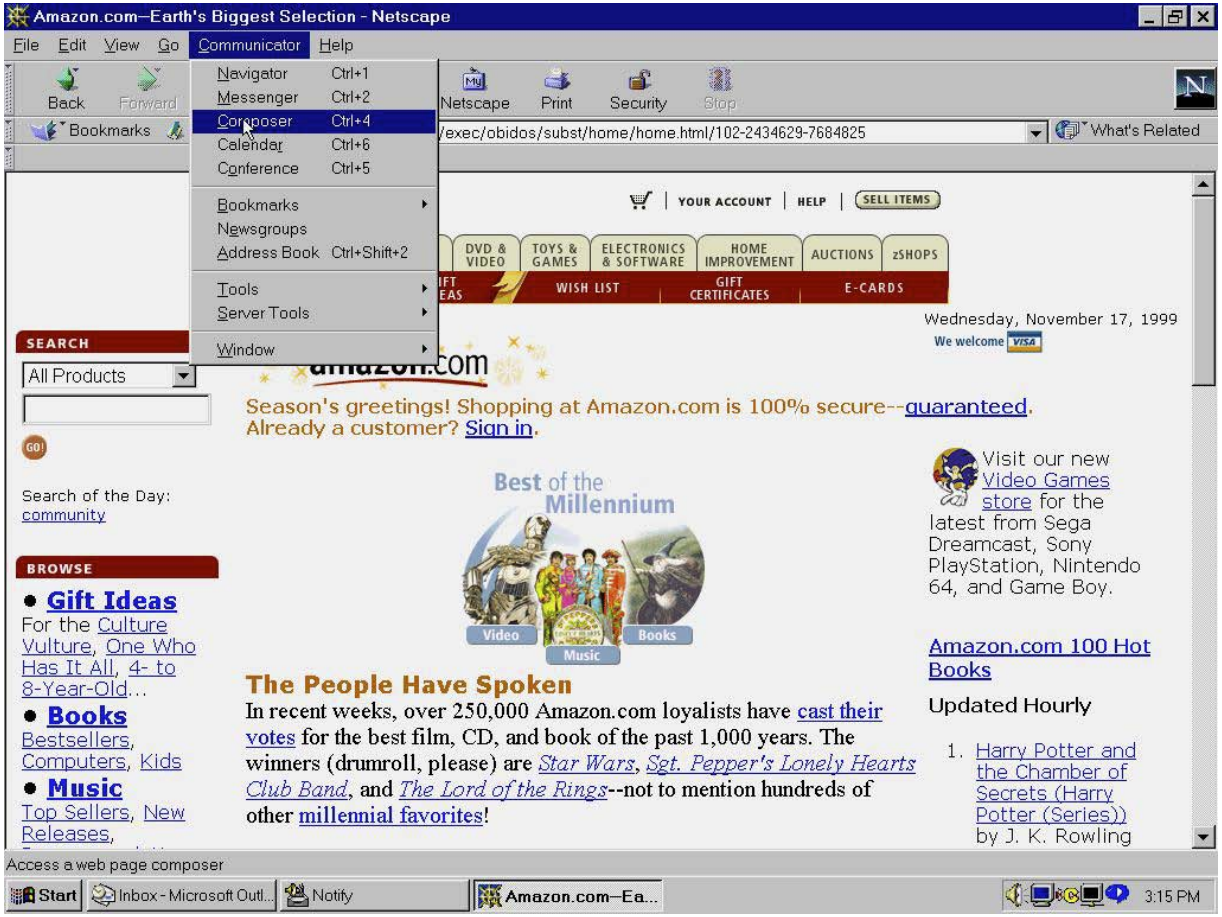


Exhibit 1.5 Examining the Major Components of the Netscape Communicator while Viewing a Popular Electronic Commerce Site

The RealPlayer provides users with the ability to listen to music and conversations or to view events in near-real-time. The term "near-real-time" is used because the player buffers data to obtain the ability to eliminate random delays associated with the arrival of packets as they flow over the Internet and encounter different degrees of delay.

Exhibit 1.6 illustrates the use of the RealNetworks' RealPlayer on the author's desktop. In this example, the author is both listening and viewing a video clip from the Fox News Network. While audio and video players can turn the desktop into miniature televisions, they can also saturate the use of bandwidth on a network. In the example shown in Exhibit 1.6, the author was watching news about the crash of an airplane when network congestion occurred, forcing the player to freeze its audio and video presentation and buffer data at a lower rate until a sufficient amount of data is buffered to allow playback. Because it is very easy for 50 to 100 employees to click on different music and news items, the cumulative effects of such actions can result in the necessity to either upgrade a network or restrict the use of audio and video players.

Voice Over IP

A second emerging application that can result in the restructuring of an existing network is the transmission of digitized voice over TCP/IP networks. Referred to as Voice over IP (VoIP), this technology is extremely delay sensitive and does not tolerate lengthy packets transporting data interspersed between packets transporting digitized voice. Thus, the ability to transmit Voice over IP can require equipment or software that prioritizes packets transporting voice over those transporting data as well as fragmenting lengthy packets transporting data, so their transmission between voice-carrying packets has a minimal effect on the reconstruction of voice at the receiver.

Virtual Private Networking

A third emerging TCP/IP application is the use of the Internet as a virtual private network (VPN). The rationale for the use of the Internet as a VPN is economics. Leased lines are billed monthly based on distance between interconnected locations and operation rate. In comparison, the use of the Internet is distance insensitive, with corporations primarily billed on a monthly basis based on the operation rate of the access line that connects each corporate location to the Internet.

In addition to reducing the cost of communications, a VPN can save on equipment costs. This is because one connection to the Internet can support an almost unlimited number of virtual paths to different geographically separated corporate locations. In comparison, a private network would require routers at each location to have multiple ports to obtain the ability to

Exhibit 1.6 Using RealNetworks' RealPlayer to Obtain the Latest News from the Fox Network

interconnect one location with many other locations. Because router ports are relatively expensive, typically costing $1000 or more each, the internetworking of a large number of organizational locations via the use of an intranet can result in the expenditure of a considerable amount of money for additional router ports. Thus, VPNs can reduce the cost of both communications hardware and transmission facilities.

While VPNs provide a mechanism to reduce networking costs, they open up networks connected to the Internet to potential attack from a virtually unlimited population of hackers. Thus, the use of VPNs introduces the need to consider various security measures to include firewalls, router access lists, and servers that perform authentication and encryption.

Given an appreciation for some of the emerging application being developed for use under the TCP/IP protocol suite, this chapter concludes with a brief preview of the focus of succeeding chapters in this book.

Book Preview

As indicated in the table of contents, this book is divided into nine chapters. Thus, this section reviews the focus of each chapter, commencing with Chapter 2, in the order by which they are incorporated into this book.

The Protocol Suite

Chapter 2 provides an introduction to the major components of the TCP/IP protocol suite. Information presented in this chapter will make us aware of the relationship of the various components of the protocol suite and their operations and functionality. By understanding the structure of the protocol suite, we will note its flexibility for adding new applications and protocols as well as how its use has evolved over the past four decades.

The Standards Process

A book covering the TCP/IP protocol suite would do disservice to its readers if it failed to provide information concerning the manner by which TCP/IP-related standards are developed. Thus, Chapter 3 focuses on this topic. Chapter 3 examines the TCP/IP standards process. Because the TCP/IP protocol suite is intimately associated with the Internet, Chapter 3 begins by the various standards-making organizations and committees connected with the Internet. Once this is accomplished, the chapter examines how certain types of documents, referred to as Requests for Comments (RFCs), evolved into standards. These allow the TCP/IP protocol suite to be continuously updated to support both advances in technology as well as changes in the communications requirements of businesses, government agencies, academia, and individual end users.

The Internet Protocol and Related Protocols

Chapter 4 commences with an examination of protocol specifics by focusing attention on the network layer. This chapter examines how the Internet Protocol (IP) operates, its addressing method, and several related protocols that are transported either by IP or at the data link layer. This resolves addressing differences between Layers 2 and 3 of the International Standards Organization (ISO) Open System Interconnection (OSI) Reference Model. The two additional protocols discussed in Chapter 4 include the Internet Control Message Protocol (ICMP) and the Address Resolution Protocol (ARP).

Transport Layer Protocols

Continuing the examination of the TCP/IP protocol suite, Chapter 5 focuses on the transport layer that represents Layer 4 of the ISO OSI Reference Model.

Chapter 5 examines the operation of the two-layer protocols included in the TCP/IP protocol suite: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). In so doing, the chapter reveals how each one operates, the similarities and differences between the protocols, and how they are used by different applications.

Applications and Built-in Diagnostic Tools

The top layer of the TCP/IP protocol suite includes a variety of applications, some of which represent diagnostic tools one can utilize to check the operation of network components. Chapter 6 examines the operation and utilization of a core set of TCP/IP applications, including several applications whose use can be considered equivalent to the use of a diagnostic tool.

Routing

Chapter 7 focuses on the manner by which packets are transmitted through a TCP/IP network. Although there are over two dozen routing protocols, attention is focused on just a few that represent protocols that route a majority of network traffic. The protocols examined include a routing protocol commonly used by small- and medium-sized networks, and a routing protocol used to interconnect autonomous networks.

Security

Because most TCP/IP networks include a connection to the Internet security, it is a most important topic and is covered as a separate chapter in this book. Chapter 8 examines the use of router access lists and firewalls as well as proxy services and network address translation.

Emerging Technologies

In concluding this book, the author examines the potential effect of four emerging technologies. First, we will describe and discuss how the new version of the Internet Protocol, IPv6, can be expected to literally open a window that will allow an extraordinary number of new devices to be connected to the Internet. Other topics to be covered in this chapter include the role of Virtual Private Networks, the growing use of Voice over IP, and Mobile IP.

Now that we have an appreciation for the orientation of this book, let us relax, grab a Coke© or a cup of coffee, and proceed to follow this author on a tour into the world of communications provided by the use of the TCP/IP protocol suite.

Chapter 2

The Protocol Suite

The primary purpose of this chapter is to obtain an appreciation for the general composition of the TCP/IP protocol suite. This can be accomplished by first examining the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model. Although the TCP/IP protocol suite predated the ISO's Reference Model, by examining the layering concept associated with communications defined by that model one can obtain a better appreciation for the functioning of the TCP/IP protocol suite.

The ISO Reference Model

During the 1970s, approximately a dozen years after the development of several popular communications protocols to include TCP/IP, the International Standards Organization (ISO) established a framework for standardizing communications systems. This framework was called the Open System Interconnection (OSI) Reference Model and defines an architecture in which communications functions are divided into seven distinct layers, with specific functions becoming the responsibility of a particular layer.

Exhibit 2.1 illustrates the seven layers of the OSI Reference Model. Note that each layer, with the exception of the lowest, covers a lower layer, effectively isolating them from higher layer functions. Layer isolation is an important aspect of the OSI Reference Model as it allows the given characteristics of one layer to change without affecting the remainder of the model, provided that support services remain the same. This is possible because of well-known interface points in a layered model that enable one layer to communicate with another although one or both may change. In addition, the layering process permits end users to mix and match OSI or other layered protocol-conforming communications products to tailor their communications system to satisfy a particular networking requirement. Thus, the OSI Reference

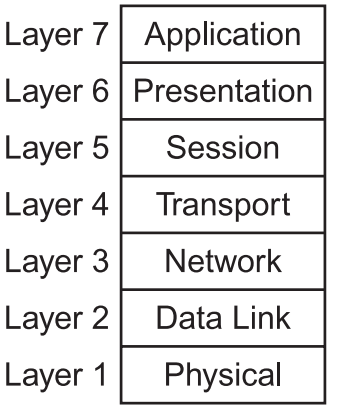


Exhibit 2.1 The ISO Open System Interconnection Reference Model

Model, as well as protocol suites that employ a layered architecture, provide the potential to directly interconnect networks based on the use of different vendor products. This architecture, which is referred to as an open architecture when its specifications are licensed or placed in the public domain, can be of substantial benefit to both users and vendors. For users, an open architecture removes them from dependence on a particular vendor, and can also prove economically advantageous as it fosters competition.

For vendors, the ability to easily interconnect their products with the products produced by other vendors opens up a wider market. Consider now the functions of the seven layers of the OSI Reference Model.

OSI Reference Model Layers

As previously noted, the OSI Reference Model consists of seven layers, with specific functions occurring at each layer. This section provides an understanding of the functions performed at each layer in the OSI Reference Model. This information can then be used in the next section in this chapter to better understand the components of the TCP/IP protocol suite.

Layer 1: The Physical Layer

The physical layer represents the lowest layer in the ISO Reference Model. Because the physical layer involves the connection of a communications system to communications media, the physical layer is responsible for specifying the electrical and physical connection between communications devices that connect to different types of media. At this layer, cable connections and the electrical rules necessary to transfer data between devices are specified. Examples of physical layer standards include RS-232, V.24, and the V.35 interface.

Layer 2: The Data Link Layer

The second layer in the ISO Reference Model is the data link layer. This layer is responsible for defining the manner by which a device gains access to the medium specified in the physical layer. In addition, the data link layer is also responsible for defining data formats to include the entity by which information is transported, error control procedures, and other link control procedures.

Most trade literature and other publications reference the entity by which information is transported at the data link layer as a frame. Depending on the protocol used, the frame will have a certain header composition with fields that normally indicate the destination address and the originator of the frame through the use of a source address. In addition, frames will have a trailer with a cyclic redundancy check (CRC) field that indicates the value of an error checking mechanism algorithm performed by the originator on the contents of the frame. The receiver will apply the same algorithm to an inbound frame and compare the locally generated CRC to the CRC in the trailer. If the two match, the frame is considered to be received without error, while a mismatch indicates a transmission error occurred, and the receiver will then request the originator to retransmit the frame. Examples or common Layer 2 protocols include such LAN protocols as Ethernet and Token Ring, as well as such WAN protocols as High Level Data Link Control (HDLC).

The original development of the OSI Reference Model targeted wide area networking. This resulted in its applicability to LANs requiring a degree of modification. The Institute of Electrical and Electronic Engineers (IEEE), which is responsible for developing LAN standards, subdivided the data link layer into two sub-layers: Logical Link Control (LLC) and Media Access Control (MAC). The LLC layer is responsible for generating and interpreting commands that control the flow of data and performing recovery operations in the event errors are detected. In comparison, the MAC layer is responsible for providing access to the local area network, which enables a station on the network to transmit information. The subdivision of the data link layer allows a common LLC layer to be used regardless of differences in the method of network access. Thus, a common LLC is used for both Ethernet and Token Ring, although their access methods are dissimilar.

Layer 3: The Network Layer

Moving up the ISO Reference Model, the third layer is the network layer. This layer is responsible for arranging a logical connection between a source and destination on the network to include the selection and management of a route for the flow of information between source and destination based on the available paths within a network.

Services or functions provided at the network layer are associated with the movement of data through a network to include addressing, routing, switching, sequencing, and flow control procedures. At the network layer, units of information are placed into packets that have a header and trailer, similar to

frames at the data link layer. Thus, the network layer packet will contain addressing information as well as a field that facilitates error detection and correction.

In a complex network, the source and destination may not be directly connected by a single path. Instead, a path may be required to be established through the network that consists of several sub-paths. Thus, the routing of packets through the network, as well as the mechanism in the form of routing protocols that provide information about available paths, are important features of this layer.

Several protocols are standardized for Layer 3 to include the International Telecommunications Union Telecommunications body (ITU-T) X.25 packet switching protocol and the ITU-TΔX.25 gateway protocol. X.25 governs the flow of information through a packet network, whereas X.75 governs the flow of information between packet networks. In examining the TCP/IP protocol suite in the next section of this chapter, one sees that the Internet Protocol (IP) represents the network layer protocol used in the TCP/IP protocol suite. One also notes that addressing at the network layer and the data link layer differ from one another, and a discovery process is used for packets to be correctly delivered via frames to their intended destination.

Layer 4: The Transport Layer

Continuing the tour of the ISO Reference Model, the transport layer is responsible for governing the transfer of information after a route has been established through the network by the network layer protocol. There are two general types of transport layer protocols: connection oriented and connectionless. A connection-oriented protocol first requires the establishment of a connection prior to data transfer occurring. This type of transport layer protocol performs error control, sequence checking, and other end-to-end data reliability functions. A second type or category of transport layer protocol operates as a connectionless, best-effort protocol. This type of protocol depends on higher layers in the protocol suite for error detection and correction. TCP in the TCP/IP protocol suite represents a Layer 4 connection-oriented protocol, while UDP represents a connectionless Layer 4 protocol.

Layer 5: The Session Layer

The fifth layer in the OSI Reference Model is the session layer. This layer is responsible for providing a set of rules that govern the establishment and termination of data streams flowing between nodes in a network. The services that the session layer can provide include establishing and terminating node connections, message flow control, dialogue control, and end-to-end data control. In the TCP/IP protocol suite, Layers 5 through 7 are grouped together as an application layer.

Layer 6: The Presentation Layer

The sixth layer of the OSI Reference Model is the presentation layer. This layer is concerned with the conversion of transmitted data into a display format appropriate for a receiving device. This conversion can include data codes as well as display placement. Other functions performed at the presentation layer can include data compression and decompression and data encryption and decryption.

Layer 7: The Application Layer

The top layer of the OSI Reference Model is the application layer. This layer functions as a window through which the application gains access to all of the services provided by the model. Examples of functions performed at the application layer include electronic mail, file transfers, resource sharing, and database access. Although the first four layers of the OSI Reference Model are fairly well defined, the top three layers can vary considerably between networks. As previously mentioned, the TCP/IP protocol suite, which is a layered protocol that predates the ISO Reference Model, combines Layers 5 through 7 into one application layer.

Data Flow

The design of an ISO Reference Model compatible network is such that a series of headers are opened to each data unit as packets are transmitted and delivered by frames. At the receiver, the headers are removed as a data unit flows up the protocol suite, until the "headerless" data unit is identical to the transmitted data unit. The next chapter section examines the flow of data in a TCP/IP network that follows the previously described ISO Reference Model data flow.

The ISO Reference Model never lived up to its intended goal, with ISO protocols achieving a less-than-anticipated level of utilization. The concept of the model made people aware of the benefits that could be obtained by a layered open architecture as well as the functions that would be performed by different layers of the model. Thus, the ISO succeeded in making networking personnel aware of the benefits that could be derived from a layered open architecture and more than likely contributed to the success of the acceptance of the TCP/IP protocol suite.

The TCP/IP Protocol Suite

The Transmission Control Protocol/Internet Protocol (TCP/IP) actually represents two distinct protocols within the TCP/IP protocol suite. Due to the popularity of those protocols, and the fact that a majority of traffic is transferred using those protocols, the members of the protocol suite include TCP and IP and are collectively referred to as TCP/IP.

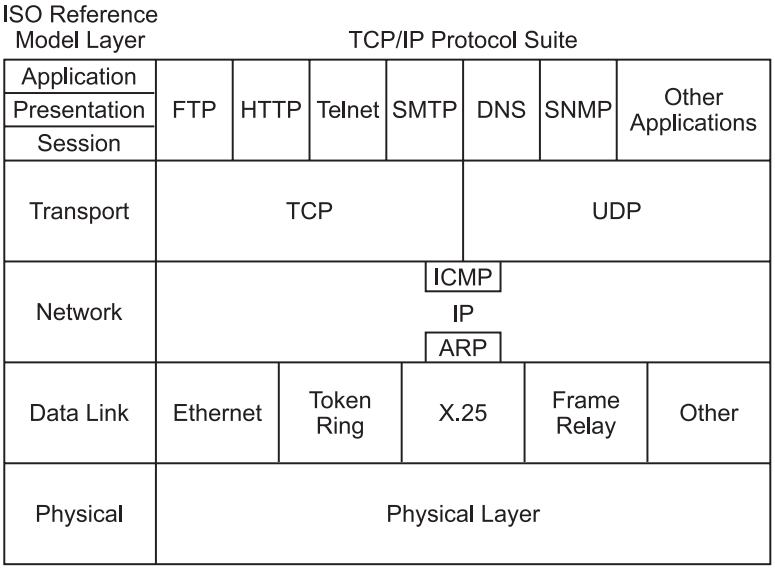


Exhibit 2.2 Comparing the TCP/IP Protocol Suite to the ISO Reference Model

Exhibit 2.2 provides a general comparison of the structure of the TCP/IP protocol suite to the OSI Reference Model. The term "general comparison" is used because the protocol suite consists of hundreds of applications, of which only a handful are shown. Another reason that Exhibit 2.2 is a general comparison results from the fact that the TCP/IP protocol suite actually begins above the data link layer. Although the physical and data link layers are not part of the TCP/IP protocol suite, they are shown in Exhibit 2.2 to provide a frame of reference to the ISO Reference Model as well as to facilitate an explanation of the role of two special protocols within the TCP/IP protocol suite.

The Network Layer

The network layer of the TCP/IP protocol stack primarily consists of the Internet Protocol (IP). The IP protocol includes an addressing scheme that identifies the source and destination address of the packet being transported. In TCP/IP terminology, the unit of data being transmitted at the network layer is referred to as a datagram. Also included in what can be considered to represent the network layer are two additional protocols that perform very critical operations. Those protocols are the Address Resolution Protocol (ARP) and the Internet Control Message Protocol (ICMP).

IP

The Internet Protocol (IP) provides the addressing capability that allows datagrams to be routed between networks. The current version of IP is IPv4,

under which IP addresses consist of 32 bits. There are currently five classes of IP addresses, referred to as Class A through Class E, with Classes A, B, and C having their 32 bits subdivided into a network portion and a host portion. The network portion of the address defines the network where a particular host resides, while the host portion of the address identifies a unique host on the network. Chapter 4 examines the Internet Protocol in detail to include its current method of 32-bit addressing. Chapter 6 focuses emerging technologies and on the next-generation Internet Protocol referred to as IPv6.

ARP

One of the more significant differences between the data link layer and the network layer is the method of addressing used at each layer. At the data link layer, such LANs as Ethernet and Token Ring networks use 48-bit MAC addresses. In comparison, TCP/IP currently uses 32-bit addresses under the current version of the IP protocol and the next generation of the IP protocol, IPv6, uses a 128-bit address. Thus, the delivery of a packet or datagram flowing at the network layer to a station on a LAN requires an address conversion. That address conversion is performed by the Address Resolution Protocol whose operation is covered in detail in Chapter 4.

ICMP

The Internet Control Message Protocol (ICMP), as its name implies, represents a protocol used to convey control messages. Such messages range in scope from routers responding to a request that cannot be honored with a "destination unreachable" message to the requestor, to messages that convey diagnostic tests and responses. An example of the latter is the echo-request/echo-response pair of ICMP datagrams that is more popularly referred to collectively as Ping.

ICMP messages are conveyed with the prefix of an IP header to the message. Thus, one can consider ICMP to represent a Layer 3 protocol in the TCP/IP protocol suite. The structure of ICMP messages as well as the use of certain messages are examined in Chapter 4 where the network layer of the TCP/IP protocol suite is examined in detail.

The Transport Layer

As indicated in Exhibit 2.2, there are two transport layer protocols supported by the TCP/IP protocol suite: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

TCP

TCP is an error-free, connection-oriented protocol. This means that prior to data being transmitted by TCP, the protocol requires the establishment of a

path between source and destination as well as an acknowledgment that the receiver is ready to receive information. Once the flow of data commences, each unit, which is referred to as a TCP segment, is checked for errors at the receiver. If an error is detected through a checksum process, the receiver will request the originator to retransmit the segment. Thus, TCP represents an error-free, connection-oriented protocol.

The advantages associated with the use of TCP as a transport protocol relate to its error-free, connection-oriented functionality. For the transmission of relatively large quantities of data or important information, it makes sense to use this transport layer protocol. The connection-oriented feature of the protocol means that it will require a period of time for the source and destination to exchange handshake information. In addition, the error-free capability of the protocol may be redundant if the higher layer in the protocol suite also performs error-checking. Recognizing the previously mentioned problems, the developers of the TCP/IP protocol suite added a second transport layer protocol referred to as UDP.

UDP

The User Datagram Protocol (UDP) is a connectionless, best effort, non-error-checking transport protocol. UDP was developed in recognition of the fact that some applications may require small pieces of information to be transferred, and the use of a connection-oriented protocol would result in a significant overhead to the transfer of data. Because a higher layer in the protocol suite could perform error-checking, error detection and correction could also be eliminated from UDP. Because UDP transmits a piece of information referred to as a UDP datagram without first establishing a connection to the receiver, the protocol is also referred to as a best-effort protocol. To ensure that a series of UDP datagrams is not transmitted into a black hole if a receiver is not available, the higher layer in the protocol suite using UDP as a transport protocol will wait for an acknowledgment. If one is not received within a predefined period of time, the application can decide whether to retransmit or cancel the session.

In examining Exhibit 2.2, note that certain applications use TCP as their transport protocol while other applications use UDP. In general, applications that require data integrity, such as remote terminal transmission (Telnet), file transfer (FTP), and electronic mail, use TCP as their transport protocol. In comparison, applications that transmit relatively short packets, such as the Domain Name Service (DNS) and the Simple Network Management Protocol (SNMP) that is used to perform network management operations use UDP.

One relatively new TCP/IP application takes advantage of both the TCP and UDP transport protocols. That application is Voice over IP (VoIP). VoIP commonly uses TCP to set up a call and convey signaling information to the distant party. Because real-time voice cannot be delayed by retransmission if an error in a packet is detected, there is no need to perform error detection.

Thus, digitized voice samples are commonly transmitted using UDP once a session is established using TCP.

Application Layer

As previously noted, the development of the TCP/IP protocol suite predated the development of the ISO's OSI Reference Model. At the time the TCP/IP protocol suite was developed, functions above the transport layer were combined into one entity that represented an application. Thus, the TCP/IP protocol suite does not include separate session and presentation layers. Now having an appreciation for the manner by which the TCP/IP protocol stack can be compared and contrasted to the OSI Reference Model, this chapter concludes by examining the flow of data within a TCP/IP network.

Data Flow

Data flow within a TCP/IP network commences at the application layer where data is provided to an applicable transport layer protocol — TCP or UDP. At Layer 4, the transport layer opens either a TCP or a UDP header to the application data, depending on the transport protocol used by the application layer.

The transport layer protocol uses a port number to distinguish the type of application data being transported. Through the use of port numbers, it becomes possible to distinguish one application from another that flow between a common source and destination.

For lay personnel not familiar with TCP or IP, this explains how a common hardware platform, such as a Windows NT server, can support both Web and FTP services. That is, although the server has a common IP address contained in an IP header, the port number in the TCP or UDP header indicates the application.

Application data flowing onto a network is first formed into a TCP segment or UDP datagram. The resulting UDP datagram or TCP segment is then passed to the network layer where an IP header is opened. The IP header contains network addressing information that is used by routers to route datagrams through a network.

When an IP datagram reaches a LAN, the difference between the network layer and LAN address is first resolved through ARP. Once this is accomplished, the IP datagram is placed into a LAN frame using an appropriate MAC address in the LAN header. Exhibit 2.3 illustrates the data flow within a TCP/IP network for delivery to a station on a LAN.

The TCP/IP protocol suite represents a methodically considered and developed collection of protocols and applications. As noted in subsequent chapters of this book, it is a very flexible open architecture that allows new applications and protocols to be developed. Concerning that development, it is the standards process that ensures the orderly development of additions to the protocol suite. Thus, Chapter 3 focuses on this important topic.

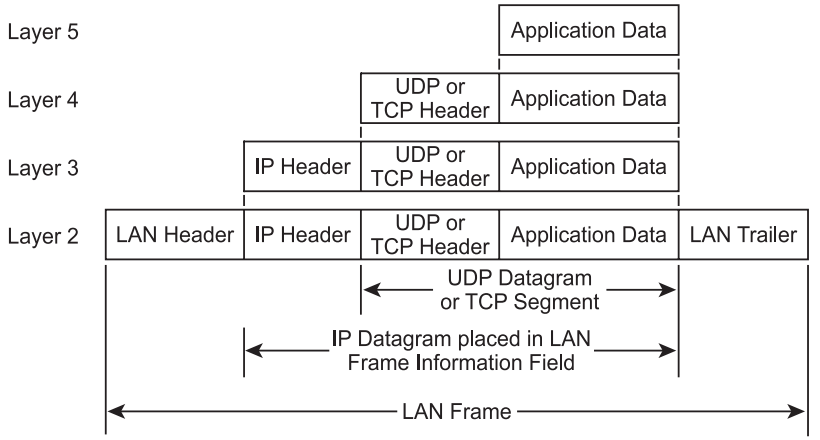


Exhibit 2.3 Data Flow within a TCP/IP Network for Delivery to a Station on a LAN

Chapter 3

Internet Governing Bodies and The Standards Process

Standards are the glue that enables hardware and software products to inter-operate. Without standards it would be difficult, if not impossible, for different vendors to create products that could operate with products made by other organizations. For the TCP/IP protocol suite, standards are developed by several organizations that can be considered as governing bodies of the Internet. Thus, the first section of this chapter concentrates on this topic. Once this is accomplished, attention turns to the standardization process and the relationship of different governing bodies to the publication of documents that affect the use of the TCP/IP protocol suite.

Internet Governing Bodies

Any discussion of the role of various bodies in governing the manner by which networks that form the Internet connect to one another and how they control the evolution of the TCP/IP protocol suite is facilitated by examining the evolution of the mother of all networks. Thus, let us digress a bit and focus attention on the manner by which funding by the U.S. Department of Defense was used to develop communications between research centers that evolved into the Internet.

Internet Evolution

The evolution of the TCP/IP protocol suite can be traced to the efforts of the U.S. Department of Defense Advanced Research Projects Agency (DARPA). During the latter portion of the 1960s, DARPA funded a project to facilitate

communications between computers that resulted in the development of a protocol referred to as the Network Control Program (NCP). For a period of approximately seven years, NCP was used to support process-to-process communications between host computers via a packet switching network operated by the Advanced Research Project Agency (ARPA) referred to as ARPAnet.

Although NCP allowed peer-to-peer communications, it lacked a degree of flexibility that resulted in DARPA providing research grants to the University of California at Los Angeles (UCLA), Stanford Research Institute (SRI), and several additional universities. This resulted in a recommendation to replace NCP with a protocol referred to as the Transmission Control Program (TCP). Between 1975 and 1979, DARPA funding resulted in the development of TCP and the protocol responsible for the routing of packets that was given the name Internet Protocol (IP). Within a short period of time, the protocol suite was referred to as TCP/IP. In 1983, ARPA required all organizations that wished to connect their computers to ARPAnet to use the TCP/IP protocol suite.

In 1983, ARPAnet was subdivided into two networks. One network, known as Military Network (MILNET), was developed for use by the Department of Defense. The second network that now represents nonmilitary sites was called the DARPA Internet.

During the mid-1980s, a large number of networks were created using the TCP/IP protocol suite. Some networks represented associations of universities within a geographical area, while other networks were developed by commercial organizations. Each of these networks was interconnected using the ARPAnet as a backbone and resulted in the beginning of what is now known as the Internet.

At the same time the ARPAnet was being used as a backbone by geographically separated regional networks, a new network was formed. The initial goal of this new network was to link five supercomputer sites. This network, operated by the National Science Foundation (NSF) and was referred to as NSFnet, was established in 1986. As a relatively new network, the NSF built a backbone with 56-Kbps circuits that were upgraded to 1.544-Mbps T1 circuits by July 1988.

Within a short period of time, several regional networks began to link their facilities to the NSFnet. Although the NSFnet was a noncommercial enterprise, several commercial networks were developed during the late 1980s that were interconnected to the NSFnet via points or locations referred to as Commercial Internet Exchanges (CIXs). Later, the CIXs evolved to become peering points that represent locations where modern-day Internet service providers interconnect their networks.

By 1989, the original ARPAnet had become expensive to operate, while NSFnet provided a faster backbone infrastructure while providing a mirror image of the ARPAnet. This resulted in DARPA deciding to take ARPAnet out of service. In turn, the use of the NSFnet further increased. As LANs became

prevalent and were connected to the NSFnet, the term "Internet" was commonly used to reference the network of interconnected networks. As the use of the Internet expanded, the NSF did not have the staff required for various administrative duties associated with running the network, and issued contracts to facilitate the orderly growth in connectivity. Some companies were given the responsibility to operate Network Access Points (NAPs) through which companies could connect commercial networks to the Internet, while other companies received contracts to register domain names, such as xyz.com and myuniversity.edu. Eventually, the NSF contracted out all of the functions associated with operation of the Internet. By 1995, the NSF shut down its backbone as the number of NAPs, which later became known as peering points, proved sufficient for network interconnection purposes and made the NSFnet obsolete. Today, there are literally thousands of Internet service providers (ISPs) whose networks are interconnected to one another through peering points. To ensure interoperability, several organizations have been formed over the past 30 years to govern various aspects of the TCP/IP protocol suite. Some of the more prominent organizations include the Internet Activities Board (IAB), which was renamed the Internet Engineering Task Force (IETF); the Internet Assigned Numbers Authority (IANA); and the Internet Society (ISOC).

The IAB and IETF

In 1983, the Internet Activities Board (IAB) was formed as an umbrella organization to coordinate the activities of independent task forces that were previously formed to focus attention on a particular area of technology, such as routing protocols, addressing, and standards. One of the working groups that gained a significant degree of prominence and a literal explosion in attendance was the Internet Engineering Task Force (IETF).

In 1992, the Internet Activities Board was both reorganized and renamed, with the new name Internet Architecture Board that allowed the same initials to be used. Today, the IAB represents a technical advisory group of the Internet Society, with the latter formed in 1991 as an umbrella organization for the IAB, IETF, and Internet Research Task Force (IRTF). The new IAB is responsible for providing oversight of the architecture for the protocols and procedures used by the Internet as well as for editorial management and publication of Request for Comments (RFC) documents and for administration of Internet assigned numbers. As noted in the second section of this chapter, RFCs are documents that define the TCP/IP protocol suite.

The IANA

The Internet Assigned Numbers Authority (IANA) until recently was supported by the U.S. government, but is now a not-for-profit organization with an

international board of directors. The IANA is responsible for Internet protocol addresses, domain names, and protocol parameters and serves as the central coordinating location for the Internet.

The IANA dates to the creation of the Internet and was originally funded by the NSF. Due to international growth, it was felt that it would be more appropriate for IANA's activities to be supported by organizations that rely upon it. Thus, the IANA converted into a new, not-for-profit organization. Its role remains the same; although the IAB is responsible for RFCs, the IANA retains responsibility for any new numbering required to identify protocols, ports, or other components of the TCP/IP protocol suite. Thus, careful coordination between the IAB and IANA is required to ensure that RFCs do not adversely impact the TCP/IP protocol suite. Given an appreciation for the governing bodies of the Internet related to the development of RFCs, one can now focus on to the manner by which the TCP/IP protocol suite is standardized by examining a Request for Comments.

Request for Comments

As noted earlier in this chapter, Request for Comments (RFCs) are documents that define the TCP/IP protocol suite. For the most part, RFCs are technical documents. However, they can cover a variety of topics to include an instruction for authors that defines the procedures for writing an RFC. There are currently over 2700 RFCs, and it was not until RFC 1543 that instructions for the author were standardized.

The Standards Process

Anyone can submit an RFC. However, the primary source of such documents is the IETF. The actual submission of an RFC begins as a memorandum that is reviewed by the Internet Engineering Steering Group (IESG) that operates under the IAB. If the memorandum is approved, the IESG sends it to an RFC editor. At this point in time, the document becomes a draft RFC.

Draft RFC

A draft RFC is considered a public document, and a peer review process occurs during which comments are received and reviewed concerning whether or not the RFC removes its draft status and is distributed as an RFC standard.

Proposed Standard and Draft Standard

An RFC is normally issued as a preliminary draft. After a period of time allowed for comments, the RFC will normally be published as a proposed standard. However, if circumstances warrant, the RFC draft can also be dropped from

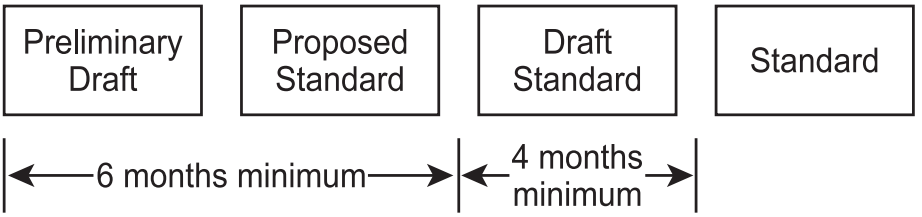


Exhibit 3.1 Internet Standards Track Time

consideration. Assuming that favorable or a lack of nonfavorable comments occur concerning the proposed standard, it can be promoted to a draft standard after a minimum period of six months.

RFC Standard

After a review period of at least four months, the Internet Engineering Steering Group (IESG) can recommend a draft standard for adoption as a standard. Although the IESG must recommend the adoption of an RFC as a standard, the IAB is responsible for the final decision concerning its adoption. Exhibit 3.1 illustrates the previously mentioned time track for the development of an RFC that represents both an Internet and TCP/IP protocol suite standard. As indicated in Exhibit 3.1, a minimum of ten months is required for an RFC to be standardized, and many times the process can require several years.

RFC Details

Once issued, an RFC is never revised; instead, an RFC is updated by new RFCs. When this situation occurs, the new RFC will indicate that it obsoletes or updates a previously published one.

RFC Categories

There are currently three categories of RFCs: Track, Informational, and Experimental. A Standards Track RFC specifies an Internet Standards Track protocol for the Internet community and requests discussion and suggestions for improvement. An Informational RFC provides information for the Internet community and does not specify an Internet Standard of any kind. The third category for RFCs is Experimental, which defines an experimental protocol for the Internet community that may or may not be adapted by the community.

Accessing RFCs

There are several locations on the Internet that maintain a repository of RFCs. Two such organizations are the RFC-Editor, a public organization, and Ohio

State University. In addition, one can also join several mailing groups to obtain RFC announcements. If one enters the keyword "RFC index" in a Web search engine, one can usually retrieve several locations where one can point the browser to access a list of RFCs. The RFC-Editor and Ohio State University operate very useful Web sites for accessing and retrieving RFCs, and probably should be considered prior to using other sites.

Exhibit 3.2 illustrates the home page of the RFC-Editor. Its Web address is http://www.rfc-editor.org/rfc.html. In examining Exhibit 3.2, note that one can search for a RFC by number, author, title, date, or keyword. In addition, one can retrieve RFCs by number and category or use the screen shown in Exhibit 3.2 to access the ability to search for and retrieve RFC.

The Computer and Information Science Department of Ohio State University operates a second Web site that warrants consideration in a search for RFCs. The Uniform Resource Locator (URL) of this site is http://www.cis.hio-state.edu/hypertext/information/rfc.html.

Exhibit 3.3 illustrates RFC-Editor and Ohio State University's support and retrieval of RFCs in a number of ways that include a keyword search. In addition, the Ohio State University Web site provides access to an Internet Users' Glossary and other documents that can be a valuable addition to anyone's "Web library."

In concluding this examination of RFC sites, it will probably be of interest to many readers to view portions of an RFC. If one selects the Index retrieval method shown in Exhibit 3.3 and scrolls down the resulting display, one notes recently published RFCs. An example of this action is shown in Exhibit 3.4, where the Ohio State University site contained 2719 online RFCs when it was accessed by this author.

In examining the entries in Exhibit 3.4, note the common format used for displaying a summary of RFCs. After the RFC number is displayed in the left margin, the title of the RFC is followed by the author's publication date, RFC format, and status. Although all RFCs must be written in 7-bit ASCII text, an approved secondary publication is in postscript. Note that by indicating the number of bytes required for storing the RFC, the index allows one to consider if one should download it via an existing connection that might not provide the bandwidth required for an expedient delivery, or if one should request its delivery via e-mail if time is not of the essence. As another option, if accessing the Index from home, one might consider waiting a return to work to access via a higher speed connection a lengthy document needed.

To illustrate the general format of an RFC, examine the relatively recent document, RFC 2710 Multicast Listener Discovery (MLD) for IPv6. From the Index listing shown in Exhibit 3.4, one sees that it is a proposed standard. By clicking on the RFC number 2710 shown in the left column in Exhibit 3.4, a display of the RFC of interest is obtained.

Exhibit 3.5 illustrates the view through a Web browser of the top portion of the beginning of the RFC. In examining Exhibit 3.5, note that the persons responsible for the RFC and their affiliations are listed at the beginning of the

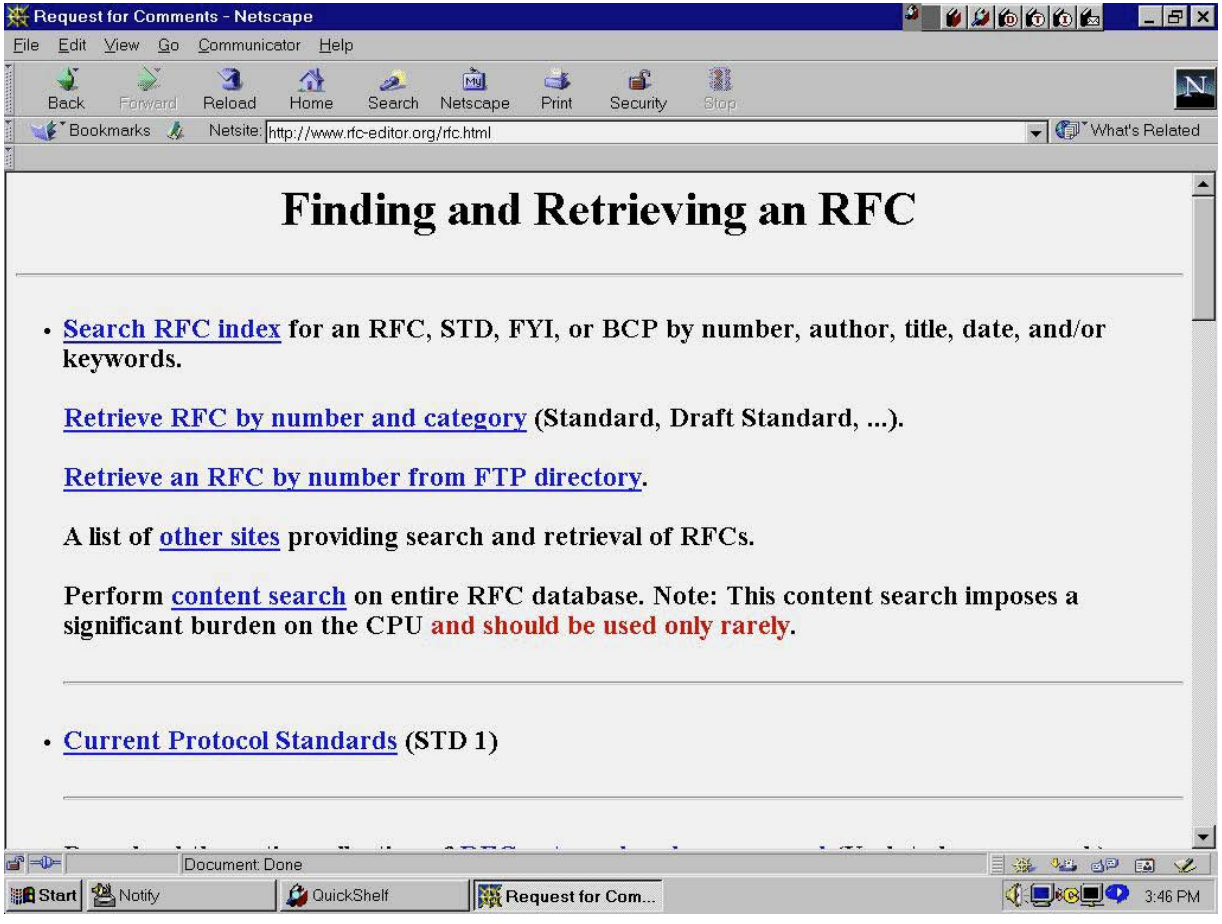

Request for Comments - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Netsite: http://www.rfc-editor.org/rfc.html What's Related
Finding and Retrieving an RFC
• Search RFC index for an RFC, STD, FYI, or BCP by number, author, title, date, and/or keywords.
Retrieve RFC by number and category (Standard, Draft Standard, ...).
Retrieve an RFC by number from FTP directory.
A list of other sites providing search and retrieval of RFCs.
Perform content search on entire RFC database. Note: This content search imposes a significant burden on the CPU and should be used only rarely.
• Current Protocol Standards (STD 1)
Document: Done
Start Notify QuickShelf Request for Com... 3:46 PM

Exhibit 3.2 The Public Rfc-Editor Provides Several Methods for Finding and Retrieving RFCs

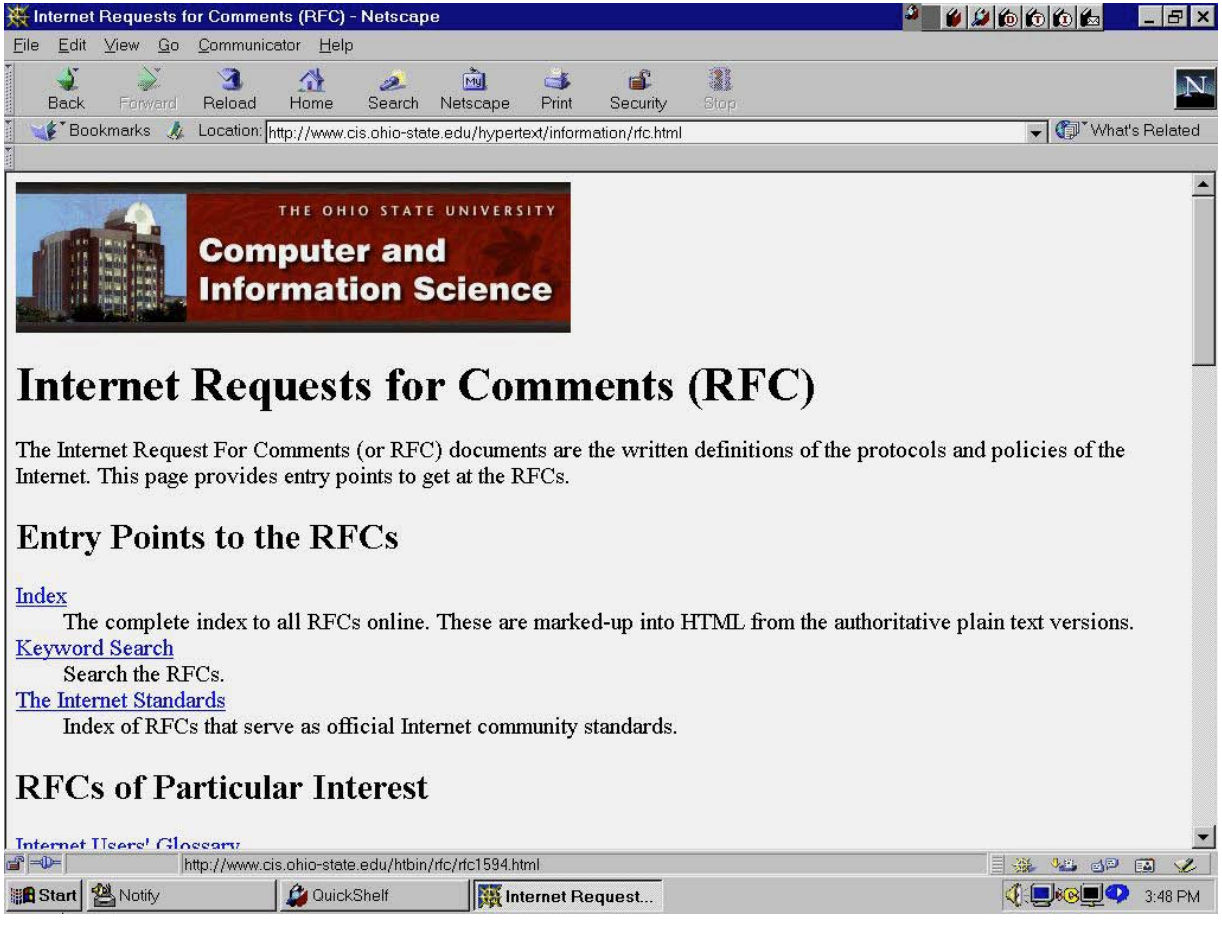

Internet Requests for Comments (RFC) - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.cis.ohio-state.edu/hypertext/information/rfc.html
What's Related
THE OHIO STATE UNIVERSITY
Computer and Information Science
Internet Requests for Comments (RFC)
The Internet Request For Comments (or RFC) documents are the written definitions of the protocols and policies of the Internet. This page provides entry points to get at the RFCs.
Entry Points to the RFCs
Index
The complete index to all RFCs online. These are marked-up into HTML from the authoritative plain text versions.
Keyword Search
Search the RFCs.
The Internet Standards
Index of RFCs that serve as official Internet community standards.
RFCs of Particular Interest
Internet Users' Glossary
http://www.cis.ohio-state.edu/htbin/rfc/rfc1594.html
Start Notify QuickShelf Internet Request... 3:48 PM

Exhibit 3.3 Viewing Access to RFCs via the Computer and Information Science Department of Ohio State University

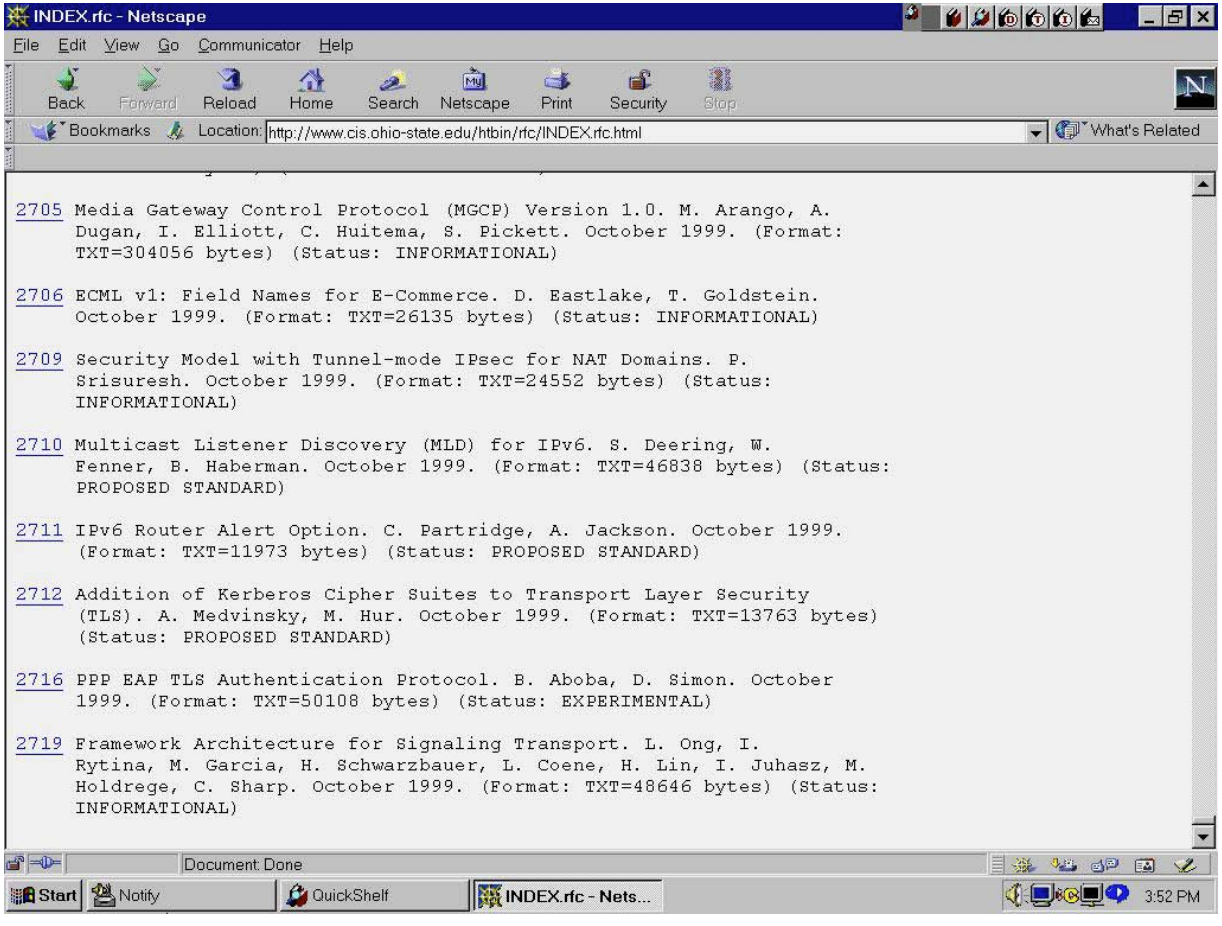

INDEX.rfc - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.cis.ohio-state.edu/htbin/rfc/INDEX.rfc.html
What's Related
2705 Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999. (Format: TXT=304056 bytes) (Status: INFORMATIONAL)
2706 ECML v1: Field Names for E-Commerce. D. Eastlake, T. Goldstein. October 1999. (Format: TXT=26135 bytes) (Status: INFORMATIONAL)
2709 Security Model with Tunnel-mode IPsec for NAT Domains. P. Srisuresh. October 1999. (Format: TXT=24552 bytes) (Status: INFORMATIONAL)
2710 Multicast Listener Discovery (MLD) for IPv6. S. Deering, W. Fenner, B. Haberman. October 1999. (Format: TXT=46838 bytes) (Status: PROPOSED STANDARD)
2711 IPv6 Router Alert Option. C. Partridge, A. Jackson. October 1999. (Format: TXT=11973 bytes) (Status: PROPOSED STANDARD)
2712 Addition of Kerberos Cipher Suites to Transport Layer Security (TLS). A. Medvinsky, M. Hur. October 1999. (Format: TXT=13763 bytes) (Status: PROPOSED STANDARD)
2716 PPP EAP TLS Authentication Protocol. B. Aboba, D. Simon. October 1999. (Format: TXT=50108 bytes) (Status: EXPERIMENTAL)
2719 Framework Architecture for Signaling Transport. L. Ong, I. Rytina, M. Garcia, H. Schwarzbauer, L. Coene, H. Lin, I. Juhasz, M. Holdrege, C. Sharp. October 1999. (Format: TXT=48646 bytes) (Status: INFORMATIONAL)
Document: Done
Start Notify QuickShelf INDEX.rfc - Nets... 3:52 PM

Exhibit 3.4 Viewing a Portion of the Ohio State University RFC Index List

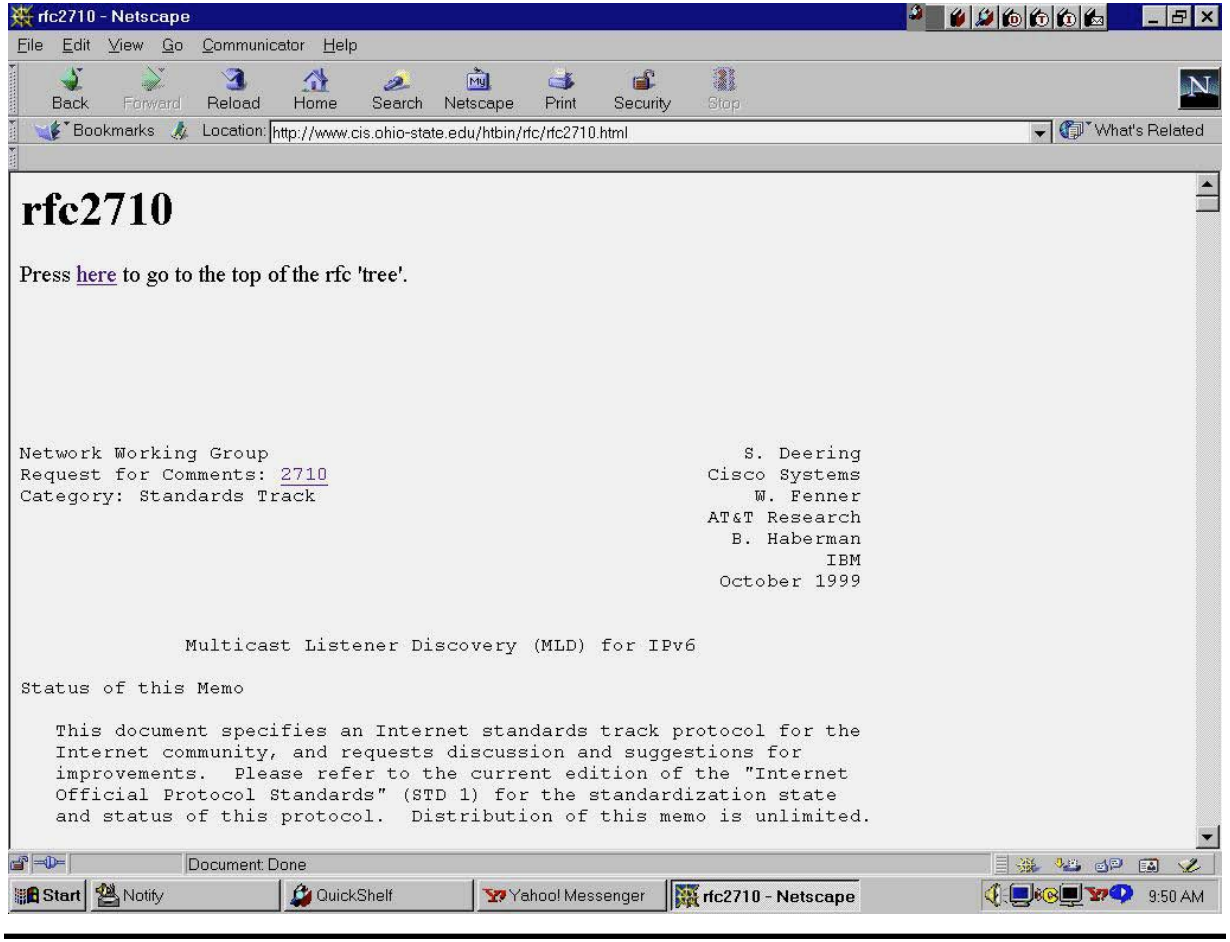

rfc2710 - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: http://www.cis.ohio-state.edu/htbin/rfc/rfc2710.html
What's Related
rfc2710
Press here to go to the top of the rfc 'tree'.
Network Working Group
Request for Comments: 2710
Category: Standards Track
S. Deering
Cisco Systems
W. Fenner
AT&T Research
B. Haberman
IBM
October 1999
Multicast Listener Discovery (MLD) for IPv6
Status of this Memo
This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.
Document: Done
Start
Notify
QuickShelf
Yahoo! Messenger
rfc2710 - Netscape
9:50 AM

Exhibit 3.5 Viewing the Initial Portion of RFC 2710

document as is the date of publication of the document. If this RFC obsoleted or updated a prior RFC, one would then note a line before the "category" line on the left side that would indicate the RFC number that was obsoleted or updated. Because the RFC viewed in Exhibit 3.5 did not obsolete or update a previously published RFC, that line was omitted from the document.

Continuing the examination of the structure of an RFC, note that the title appears on a line below the submission date. Under the title is a status section that contains a paragraph that describes the RFC. Each RFC must include on its first page a "Status of this Memo" section, which functions as a brief introduction to the RFC. Here, the term "Memo" is used, in actuality, as a memo following a format and structure evolves into an RFC.

In continuing to view the contents of an RFC, one will encounter a copyright notice, followed by an abstract of the document. Some documents will also include a table of contents, followed by the body of the document. Most modern RFCs now terminate with three sections. The second from the last section contains a section titled Author's Addresses, which lists the authors, their organization, mailing address, telephone number, and e-mail address. The next to last section in the RFC contains a complete copyright notice. The last section in an RFC contains an acknowledgment section. Thus, the basic RFC provides information on how to contact the authors as well as a detailed description of the technology it is defining.

Chapter 4

The Internet Protocol and Related Protocols

The focus of this chapter is on the first layer of the TCP/IP protocol suite. While the Internet Protocol (IP) is the primary protocol most people associate with the network layer, there are two related protocols that must be considered when discussing the TCP/IP protocol suite. Those protocols are the Address Resolution Protocol (ARP) and the Internet Control Message Protocol (ICMP).

This chapter focuses attention on what this author commonly refers to as the Network Layer Troika of the TCP/IP protocol suite: IP, ARP, and ICMP. In examining the Internet Protocol, pay particular attention to the structure of the IP header and its fields, which are examined by routers as a mechanism for making forwarding decisions. Another specific IP area of focus is addressing, as the composition of IP addresses determines the manner by which datagrams are routed from source to destination, as well as the number of hosts that can be connected to a specific type of network. In addition, in examining IP addressing, this chapter also discusses several little-known areas of the IP protocol that having knowledge about can provide the user with network design and operation flexibility. Two examples of such topics are the assignment of multiple network addresses to an interface and the use of a zero subnet. Because the filtering of IP datagrams by routers and firewalls can occur based on IP addresses, as well as ICMP message types and control field values, the information presented in this chapter will also provide a firm foundation for discussion of security in a later chapter in this book.

The initial focus in this chapter is on the IP protocol, to include its use for routing datagrams across a network and between interconnected networks. The composition of the IP header and the use of different fields in the header are examined in detail. Once this is accomplished, attention turns to the role

and operation of the Address Resolution Protocol (ARP) and includes examining the rationale for a little-known ARP technique that can considerably facilitate the operation of delay-sensitive transmissions, such as Voice over IP. A discussion of Internet Control Message Protocol (ICMP) concludes this chapter. Because some ICMP types of messages are commonly used by hackers as a mechanism to begin an attack upon a network, information about ICMP presented in this chapter will be used when examining security as a separate entity later in this book.

The Internet Protocol

The Internet Protocol (IP) represents the network layer of the TCP/IP protocol suite. IP was developed as a mechanism to interconnect packet-switched TCP/IP-based networks to form an internet. Here, the term "internet" with a lower case "i" is used to represent the connection of two or more TCP/IP-based networks.

Datagrams and Segments

The Internet Protocol transmits blocks of data referred to as datagrams. As indicated in Chapter 2, IP receives upper layer protocol data containing either a TCP or UDP header, referred to as a TCP segment or UDP datagram. The prefix of an IP header to the TCP segment or UDP datagram results in the formation of an IP datagram. This datagram contains a destination IP address that is used for routing purposes.

Datagrams and Datagram Transmission

To alleviate potential confusion between datagrams and an obsolete transmission method referred to as datagram transmission, a few words are in order. When the ARPAnet evolved, two methods of packet transmission were experimented with. One method was referred to as datagram transmission and avoided the use of routers to perform table lookups. Under datagram transmission, each node in a network transmits a received datagram onto all ports other than the port the datagram was received on. While this technique avoids the need for routing table lookup operations, it can result in duplicate datagrams being received at certain points within a network. This results in the necessity to develop software to discard duplicate datagrams, adding an additional level of complexity to networking. Thus, datagram transmission was soon discarded in favor of the creation of virtual circuits that represent a temporary path established between source and destination. When referring to datagram transmission in the remainder of this book, one is actually referencing the transmission of datagrams via a virtual circuit created between source and destination.

Routing

The actual routing of an IP datagram occurs on a best-effort or connectionless delivery mechanism. This is because IP by itself does not establish a session between the source and destination before it transports datagrams. When IP transports a TCP segment, the TCP header results in a connection-oriented session between two Layer 4 nodes transported by IP as a Layer 3 network protocol.

The importance of IP can be noted by the fact that routing between networks is based on IP addresses. As noted later in this chapter, the device that routes data between different IP addressed networks is known as a router. Because it would be extremely difficult, if not impossible, to statically configure every router in a large network to know the route to other routers and networks connected to different routers, routing protocols are indispensable to the operation of a dynamic series of interconnected IP networks. Thus, information presented in this chapter will also form a foundation for understanding the use of routing protocols, which covered as a separate entity in a later chapter of this book. The best way to obtain an appreciation for the operation of the Internet Protocol is through an examination of the fields in its header.

The IP Header

The current version of the Internet Protocol is version 4, resulting in IP commonly referred to as IPv4. The next generation of the Internet Protocol is IPv6. This section focuses attention on IPv4; IPv6 is discussed in the chapter that examines evolving technologies (Chapter 9).

Exhibit 4.1 illustrates the fields contained in the IPv4 header. In examining the IPv4 header in Exhibit 4.1, note that the header consists of a minimum of 20 bytes of data, with the width of each field shown with respect to a 32-bit (four-byte) word.

Bytes Versus Octets

In this book, the term "byte" is used to reference a sequence of eight bits used in a common manner. During the development of the TCP/IP protocol

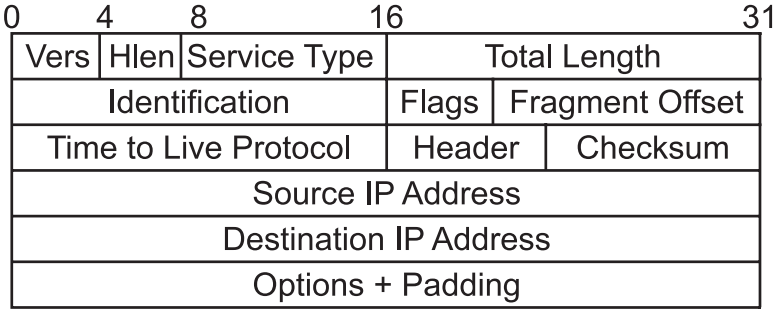


Exhibit 4.1 The Ipv4 Header

suite and continuing today, most standards documents use the term "octet" to reference a collection of eight bits. The use of the term "octet" is due to differences in the composition of a byte during the 1960s.

During the early development of computer systems, differences in computer architecture resulted in the use of groupings of five to ten bits to represent a computer byte. Thus, the term "byte" at that time was ambiguous, and standards-making bodies decided to use the term "octet" to reference a grouping of eight bits. Because all modern computers use eight-bit bytes, the term "byte" is no longer ambiguous. Thus, the term "byte" is used throughout this book.

To obtain an appreciation for the operation of IP, examine the functions of the fields in the header. When appropriate, there is discussion of the relation of certain fields to routing and security, topics that will be discussed in detail in later chapters.

Vers Field

The Vers field is four bits in length and is used to identify the version of the IP protocol used to create an IP datagram. The current version of IP is v4, with the next generation of IP assigned version number 6.

The four bits in the Vers field support 16 version numbers. Under RFC 1700, a listing of Internet version numbers can be obtained and a summary of that listing is included in Exhibit 4.2. In examining Exhibit 4.2, note that the reason the next-generation Internet Protocol is IPv6 instead of IPv5 relates to the fact that version 5 was previously assigned to an experimental protocol referred to as the Streams 2 Protocol.

Hlen Field

The length of the IP header can vary due to its ability to support options. To allow a receiving device to correctly interpret the contents of the header from

Exhibit 4.2 Assigned Internet Version Numbers

Numbers	*Assignment*
0	Reserved
1–3	Unassigned
4	IP
5	Streams
6	IPv6
7	TP/IX
8	P Internet Protocol (PIP)
9	TUBA
10–14	Unassigned
15	Reserved

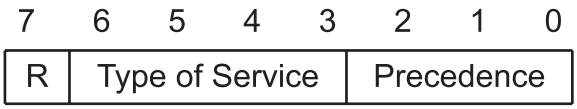
the rest of an IP datagram requires the receiving device to know where the header ends. This function is performed by the Hlen field, the value of which indicates the length of the header.

The Hlen field is four bits in length. In Exhibit 4.1, note that the IP header consists of 20 bytes of fixed information followed by options. Because it is not possible to use a four-bit field to directly indicate the length of a header equal to or exceeding 320 bytes, the value in this field represents the number of 32-bit words in the header. For example, the shortest IP header is 20 bytes, which represents 160 bits. When divided by 32 bits, this results in a value of 160/32 or 5, which is the value set into the Hlen field when the IP header contains 20 bytes and no options.

Service Type Field

The Service Type field is an eight-bit field that is commonly referred to as a Type of Service (ToS) field. The initial development of IP assumed that applications would use this field to indicate the type of routing path they would like. Routers along the path of a datagram would examine the contents of the Service Type byte and attempt to comply with the setting in this field.

Exhibit 4.3 illustrates the format of the Service Type field. This field consists of two sub-fields: Type of Service (ToS) and Precedence. The Type of Service sub-field consists of bit positions that, when set, indicate how a datagram should be handled. The three bits in the Precedence sub-field allow the transmitting station to indicate to the IP layer the priority for sending a datagram.



where:

R represents reserved

Precedence provides eight levels, 0 to 7, with 0 normal and 7 the highest.

Type of Service (ToS) indicates how the datagram is handled.

0000 Default
0001 Minimize monetary cost
0010 Maximize reliability
0100 Maximize throughput
1000 Minimize delay
1111 Maximize security

Exhibit 4.3 The Service Type Field

A value of 000 indicates a normal precedence, while a value of 111 indicates the highest level of precedence and is normally used for network control.

The value in the Precedence sub-field is combined with a setting in the Type of Service sub-field to indicate how a datagram should be processed. As indicated in the lower portion of Exhibit 4.3, there are six settings defined for the Type of Service sub-field. To understand how this sub-field is used, assume an application is transmitting digitized voice that requires minimal routing delays due to the effect of latency on the reconstruction of digitized voice. By setting the Type of Service sub-field to a value of 1000, this would indicate to each router in the path between source and destination network that the datagram is delay sensitive and its processing by the router should minimize delay.

In comparison, because routers are designed to discard packets under periods of congestion, an application in which the ability of packets to reach their destination is of primary importance would set the ToS sub-field to a value of 0010. This setting would denote to routers in the transmission path that the datagram requires maximum reliability. Thus, routers would select other packets for discard prior to discarding a packet with its ToS sub-field set to a value of 0010.

Although the concept behind including a Service Type field was a good idea, from a practical standpoint it is rarely used. The reason for its lack of use is the need for routers supporting this field to construct and maintain multiple routing tables. While this is not a problem for small networks, the creation and support of multiple routing tables can significantly affect the level of performance of routers in a complex network such as the Internet.

Although routers in most networks ignore the contents of the Service Type field, this field is now being used to map IP datagrams being transmitted over an ATM backbone. Because ATM includes a built-in Quality of Service (QoS) that, at the present time, cannot be obtained on an IP network, many organizations are transmitting a variety of data to include Voice over IP over an ATM backbone, using the Service Type field as a mechanism to map different IP service requirements into applicable types of ATM service. A second emerging application for the Service Type field is to differentiate the requirements of different applications as they flow into an IP network. In this situation, the Service Type byte is renamed as the DiffServe (Differentiated Service) byte. The Internet Engineering Task Force is currently examining the potential use of the DiffServe byte as a mechanism to define an end-to-end QoS capability through an IP network.

Total Length Field

The Total Length field indicates the total length of an IP datagram in bytes. This length indicates the length of the IP header to include options, followed by a TCP or UDP header or another type of header, as well as the data that follows that header. The Total Length field is 16 bits in length, resulting in an IP datagam having a maximum defined length of 2^{16} or 65,535 bytes.

Identification and Fragment Offset Fields

Unlike some types of clothing where one size fits all, an IP datagram can range up to 65,535 bytes in length. Because some networks only support a transport frame that can carry a small portion of the theoretical maximum-length IP datagram, it can become necessary to fragment the datagram for transmission between networks. One example of this would be the routing of a datagram from a Token Ring network to another Token Ring network via an Ethernet network. Token Ring networks that operate at 16 Mbps can transport approximately 18 Kbytes in their Information field. In comparison, an Ethernet frame has a maximum-length Information field of 1500 bytes. This means that datagrams routed between Token Ring networks via an Ethernet network must be subdivided, or fragmented, into a maximum length of 1500 bytes for Ethernet to be able to transport the data.

The default IP datagram length is referred to as the path MTU (or maximum transmission unit). The MTU is defined as the size of the largest packet that can be transmitted or received through a logical interface. For the previous example of two Token Ring networks connected via an Ethernet network, the MTU would be 1500 bytes. Because it is important to commence transmission with the lowest common denominator packet size that can flow through different networks, and, if possible, adjust the packet size after the initial packet reaches its destination, IP datagrams use a default of 576 bytes when datagrams are transmitted remotely (off the current network).

Fragmentation is a most interesting function as it allows networks capable of transmitting larger packets to do so more efficiently. The reason efficiency increases is due to the fact that larger packets have proportionally less overhead. Unfortunately, the gain in packet efficiency is not without cost. First, although routers can fragment datagrams, they do not reassemble them, leaving it to the host to perform reassembly. This is because router CPU and memory requirements would considerably expand if they had to reassemble datagrams flowing to networks containing hundreds or thousands of hosts. Second, although fragmentation is a good idea for boosting transmission efficiency, a setting in the Flag field (see below) can be used to indicate that a datagram should not be fragmented. Because many routers do not support fragmentation, many applications by default set the do not fragment flag bit and use a datagram length that, while perhaps not most efficient, ensures that a datagram can flow end-to-end as its length represents the lowest common denominator of the networks it will traverse.

When an IP datagram is fragmented, this situation results in the use of three fields in the IP header. Those fields are Identification, Flags, and Fragment Offset.

The Identification field is 16 bytes in length and is used to indicate which datagram fragments belong together. A receiving device operation at the IP network layer uses the Identification field as well as the source IP address to determine which fragments belong together. To ensure fragments are put back together in their appropriate order requires a mechanism to distinguish one fragment from another. That mechanism is provided by the Fragment Offset field, which indicates the location where each fragment belongs in a complete

message. The actual value in the Fragment Offset field is an integer that corresponds to a unit of eight bytes that indicates the offset from the previous datagram. For example, if the first fragment were 512 bytes in length, the second fragment would have an offset value that indicates that this IP datagram commences at byte 513. By using the Total Length and Fragment Offset fields, a receiver can easily reconstruct a fragmented datagram.

Flag Field

The third field in the IP header directly associated with fragmentation is the Flag field. This field is four bytes in length, with two bits used to denote fragmentation information. The setting of one of those bits is used as a direct fragment control mechanism, because a value of "0" indicates the datagram can be fragmented, while a value of "1" indicates do not fragment the datagram. The second fragment bit is used to indicate fragmentation progress. When the second bit is set to a value of "0," it indicates that the current fragment in a datagram is the last fragment. In comparison, a value of "1" in this bit position indicates that more fragments follow.

Time to Live Field

The Time to Live (TTL) field is eight bits in length. The setting in this field is used to specify the maximum amount of time that a datagram can exist. It is used to prevent a mis-addressed datagram from endlessly wandering the Internet or a private IP network, similar to the manner by which a famous American folk hero was noted in a song to wander the streets of Boston.

Because an exact time is difficult to measure, the value placed into the TTL field is actually a router hop count. That is, routers decrement the value of the TTL field by 1 as a datagram flows between networks. If the value of this field reaches zero, the router will discard the datagram and, depending on the configuration of the router, generate an ICMP message that informs the originator of the datagram that the TTL field expired and the datagram, in effect, was sent to the great bit bucket in the sky.

Many applications set the TTL field value to default of 32, which should be more than sufficient to reach most destinations in a very complex network, to include the Internet. In fact, one popular application referred to as traceroute will issue a sequence of datagrams commencing with a value of 1 in the TTL field to obtain a sequence of router-generated ICMP messages that enables the path from source to destination to be noted. The operation of the traceroute application and how it can be used as a diagnostic tool are examined in Chapter 6.

Protocol Field

It was noted in Chapter 2 that an IP header prefixes the transport layer header to form an IP datagram. While TCP and UDP represent a large majority of Layer 4 protocols carried in an IP datagram, they are not the only protocols

transported. In addition, even if they were, one would need a mechanism to distinguish one upper layer protocol from another that is carried in a datagram.

The method used to distinguish the upper layer protocol carried in an IP datagram is obtained through the use of a value in the Protocol field. For example, a value of decimal 6 is used to indicate that a TCP header follows the IP header, while a value of decimal 17 indicates that a UDP header follows the IP header in a datagram.

The Protocol field is eight bits in length, permitting up to 256 protocols to be defined under IPv4. Exhibit 4.4 lists the current assignments of Internet protocol numbers. Note that although TCP and UDP by far represent the vast majority of TCP/IP traffic on the Internet and corporate intranets, other protocols can be transported and a large block of protocol numbers are currently unassigned. Also note that under IPv6, the protocol field is named the Next Header field. Chapter 9 examines IPv6 in detail.

Header Checksum Field

The Header Checksum field contains a 16-bit cyclic redundancy check (CRC) character. The CRC represents a number generated by treating the data in the IP header field as a long binary number and dividing that number by a fixed polynomial. The result of this operation is a quotient and remainder, with the remainder being placed into the 16-bit Checksum field by the transmitting device. When a receiving station reads the header, it also performs a CRC operation on the received data, using the same fixed polynomial. If the computed CRC does not match the value of the CRC in the Header Checksum field, the receiver assumes the header is in error and the packet is discarded. Thus, the Header Checksum, as its name implies, provides a mechanism for ensuring the integrity of the IP header.

Source and Destination Address Fields

Both the Source and Destination Address fields are 32 bits in length under IPv4. The Source Address represents the originator of the datagram, while the Destination Address represents the recipient.

Under IPv4, there are five classes of IP addresses, referred to as Class A through Class E. Classes A, B, and C are subdivided into a network portion and a host portion and represent addresses used on the Internet and private IP-based networks. Classes D and E represent two special types of IPv4 network addresses. Because it is extremely important to understand the composition and formation of IP addresses to correctly configure devices connected to an IP network, as well as to design and modify such networks, the next section in this chapter focuses on this topic. Given an appreciation of IP addressing, one can then examine the use of the Address Resolution Protocol (ARP), noting how ARP is used to enable Layer 3 IP datagrams that use 32-bit IP addresses to be correctly delivered by LANs using 48-bit Layer 2 MAC addresses.

Exhibit 4.4 Assigned Internet Protocol Numbers

Decimal	*Keyword*	*Protocol*
0	HOPOPT	IPv6 Hop-by-Hop Option
1	ICMP	Internet Control Message Protocol
2	IGMP	Internet Group Management Protocol
3	GGP	Gateway-to-Gateway Protocol
4	IP	IP in IP (encapsulation)
5	ST	Stream
6	TCP	Transmission Control Protocol
7	CBT	CBT
8	EGP	Exterior Gateway Protocol
9	IGP	Any private interior gateway (used by Cisco for their IGRP)
10	BBN-RCC-MON	BBN RCC Monitoring
11	NVP-II	Network Voice Protocol, Version 2
12	PUP	PUP
13	ARGUS	ARGUS
14	EMCON	EMCON
15	XNET	Cross Net Debugger
16	CHAOS	Chaos
17	UDP	User Datagram Protocol
18	MUX	Multiplexing
19	DCN-MEAS	DCN Measurement Subsystems
20	HMP	Host Monitoring Protocol
21	PRM	Packet Radio Measurement
22	XNS-IDP	XEROX NS IDP
23	TRUNK-1	Trunk-1
24	TRUNK-2	Trunk-2
25	LEAF-1	Leaf-1
26	LEAF-2	Leaf-2
27	RDP	Reliable Data Protocol
28	IRTP	Internet Reliable Transaction Protocol
29	ISO-TP4	ISO Transport Protocol Class 4
30	NETBLT	Bulk Data Transfer Protocol
31	MFE-NSP	MFE Network Services Protocol
32	MERIT-INP	MERIT Internodal Protocol
33	SEP	Sequential Exchange Protocol
34	3PC	Third Party Connect Protocol
35	IDPR	Inter-Domain Policy Routing Protocol
36	XTP	XTP
37	DDP	Datagram Delivery Protocol
38	IDPR-CMTP	IDPR Control Message Transport Protocol
39	TP++	TP++ Transport Protocol
40	IL	IL Transport Protocol
41	IPv6	IPv6

Exhibit 4.4 Assigned Internet Protocol Numbers (continued)

Decimal	*Keyword*	*Protocol*
42	SDRP	Source Demand Routing Protocol
43	IPv6-Route	Routing Header for IPv6
44	IPv6-Frag	Fragment Header for IPv6
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Reservation Protocol
47	GRE	General Routing Encapsulation
48	MHRP	Mobile Host Routing Protocol
49	BNA	BNA
50	ESP	Encap Security Payload for IPv6
51	AH	Authentication Header for IPv6
52	I-NLSP	Integrated Net Layer Security
53	SWIPE	IP with Encryption
54	NARP	NBMA Address Resolution Protocol
55	MOBILE	IP Mobility
56	TLSP	Transport Layer Security Protocol (using Kryptonet key management)
57	SKIP	SKIP
58	IPv6-ICMP	ICMP for IPv6
59	IPv6-NoNxt	No Next Header for IPv6
60	IPv6-Opts	Destination Options for IPv6
61		Any host internal protocol
62	CFTP	CFTP
63		Any local network
64	SAT-EXPAK	SATNET and Backroom EXPAK
65	KRYPTOLAN	Kryptolan
66	RVD	MIT Remote Virtual Disk Protocol
67	IPPC	Internet Pluribus Packet Core
68		Any distributed file system
69	SAT-MON	SATNET Monitoring
70	VISA	VISA Protocol
71	IPCV	Internet Packet Core Utility
72	CPNX	Computer Protocol Network Executive
73	CPHB	Computer Protocol Heart Beat
74	WSN	Wang Span Network
75	PVP	Packet Video Protocol
76	BR-SAT-MON	Backroom SATNET Monitoring
77	SUN-ND	SUN ND PROTOCOL-Temporary
78	WB-MON	WIDEBAND Monitoring
79	WB-EXPAK	WIDEBAND EXPAK
80	ISO-IP	ISO Internet Protocol
81	VMTP	VMTP
82	SECURE-VMTP	SECURE-VMPT
83	VINES	VINES

(continues)

Exhibit 4.4 Assigned Internet Protocol Numbers (continued)

Decimal	*Keyword*	*Protocol*
84	TTP	TTP
85	NSFNET-IGP	NSFNET-IGP
86	DGP	Dissimilar Gateway Protocol
87	TCF	TCF
88	EIGRP	EIGRP
89	OSPFIGP	OSPFIGP
90	Sprite-RPC	Sprite RPC Protocol
91	LARP	Locus Address Resolution Protocol
92	MTP	Multicast Transport Protocol
93	AX.25	AX.25 Frames
94	IPIP	IP-within-IP Encapsulation Protocol
95	MICP	Mobile Internetworking Control Protocol
96	SCC-SP	Semaphore Communications Sec. Protocol
97	ETHERIP	Ethernet-within-IP Encapsulation
98	ENCAP	Encapsulation Header
99		Any private encryption scheme
100	GMTP	GMTP
101	IFMP	Ipsilon Flow Management Protocol
102	PNNI	PNNI over IP
103	PIM	Protocol Independent Multicast
104	ARIS	ARIS
105	SCPS	SCPS
106	QNX	QNX
107	A/N	Active Networks
108	IPPCP	IP Payload Compression Protocol
109	SNP	Sitara Networks Protocol
110	Compaq-Peer	Compaq Peer Protocol
111	IPX-in-IP	IPX in IP
112	VRRP	Virtual Router Redundancy Protocol
113	PGM	PGM Reliable Transport Protocol
114		Any 0-hop protocol
115	L2TP	Layer-Two Tunneling Protocol
116	DDX	D-II Data Exchange (DDX)
117–254		Unassigned
255	Reserved	

IP Addressing

This section focuses on the mechanism that enables IP datagrams to be delivered to unique or predefined groups of hosts. That mechanism is the addressing method used by the Internet Protocol, commonly referred to as IP addressing.

Under the current version of the Internet Protocol, IPv4, 32-bit binary numbers are used to identify the source and destination address in each datagram. It was not until RFC 760 that the Internet Protocol as we know it was defined and the next IP-related RFC, RFC 791 that obsoleted RFC 760, included the concept of IP address classes. Another key IP-related addressing RFC is RFC 950, which introduced the concept of subnetting. Subnetting represents a method of conserving IP network addresses and is described and discussed in detail later in this section.

Overview

Although a host is normally associated with a distinct IP address, in actuality IP addresses are used by the Internet Protocol to identify distinct device interfaces. That is, each interface on a device has a unique IP address. This explains how a router with multiple interfaces can receive communications addressed to the device on different router ports connected to LANs and WANs. Devices such as hosts, routers, and gateways can have either single or multiple interfaces. When the latter situation occurs, the device will be assigned multiple IP addresses — one for each interface.

Because most hosts are connected to a LAN via a single interface, most readers familiar with IP addressing associate a single IP address with a host. Although not as common as host workstations that use a single network connection, some servers and all firewalls and routers have multiple network connections. Exhibit 4.5 illustrates a network structure used to connect a corporate private network to the Internet. In this example, a demilitarized (DMZ)

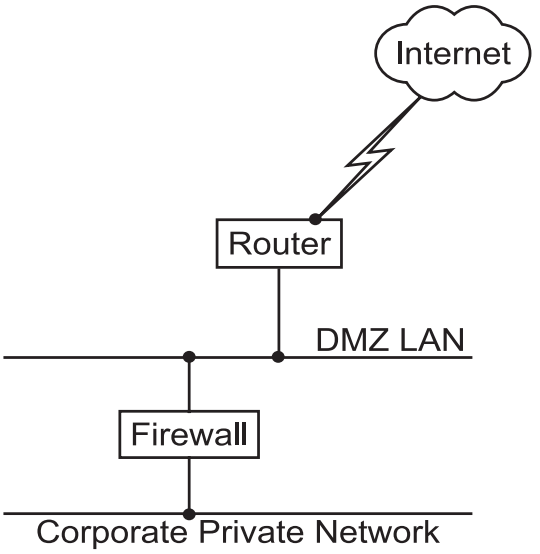


Exhibit 4.5 Several Types of Communications Devices with Multiple Interfaces, with an IP Address Assigned to Each Interface

LAN is used to interconnect the router and firewall. A DMZ LAN is a LAN without servers or workstations, in effect forcing all communications to and from the Internet to pass through a firewall. Note that both the router and firewall have multiple ports. Thus, in an IP networking environment, each communications device would be assigned two IP addresses: one for each device interface.

The IP Addressing Scheme

As previously mentioned, IPv4 uses 32-bit binary numbers to identify the source and destination address in each datagram. The use of 32-bit numbers provides an address space that supports 2^{32} or 2,294,967,296 distinct addressable interfaces. While this number probably exceeded the world's population when the Internet was initially developed as a mechanism to interconnect research laboratories and universities, the proliferation of personal computers and the development of the Web significantly expanded the role of the "mother of all networks." Recognizing that many individuals would eventually use Personnel Digital Assistants (PDAs), and even all phones to access the Web, as well as the fact that hundreds of millions in the Third World would eventually be connected to the Internet, it became obvious that IP address space would eventually be depleted. In 1992, the Internet Architecture Board (IAB) began work on a replacement for the current version of IP. Although its efforts were primarily concerned with the addressing limitations of IPv4, the IAB also examined the structure of IP and the inability of the current version of the protocol to easily indicate different options within the header. The result of the IAB effort was a new version of IP that is referred to as IPv6. IPv6 was finalized during 1995 and is currently being evaluated on an experimental portion of the Internet. Under IPv6, source and destination addresses were expanded to 128 bits, and the IP header was considerably altered, with only the Ver field retaining its position in the IPv6 header. Although the use of IPv6 will considerably enhance the support of an expanded Internet as well as facilitate various routing operations, it will be many years before the new protocol moves from an experimental status into production. Due to this, the focus on addressing in this section is on IPv4, and coverage of IPv6 is deferred to Chapter 9.

Address Changes

During the development of the Internet Protocol, it was recognized that hosts would be connected to different networks and that those networks could be interconnected to one another to form a network of interconnected networks, now commonly referred to as the Internet. Thus, in developing an IP addressing scheme, it was also recognized that a mechanism would be required to identify a network as well as a host connected to a network. This recognition resulted in the development of an addressing scheme in which certain classes of IP addresses are subdivided into a two-level addressing hierarchy.

Under the two-level IP addressing hierarchy, the 32-bit IP address is subdivided into network and host portions. The composition of the first four bits of the 32-bit word specifies whether the network portion is 1, 2, or 3 bytes in length, resulting in the host portion being either 3, 2, or 1 bytes in length.

Exhibit 4.6 The Two-Level IP Addressing Hierarchy used for Class A, B, and C Addresses

Exhibit 4.6 illustrates the two-level addressing hierarchy used by Class A, B, and C addresses whose composition and utilization are reviewed below. In examining the two-level IP addressing scheme shown in Exhibit 4.6, it should be noted that all hosts on the same network are usually assigned the same network prefix, but must have a unique host address to differentiate one host from another. As noted later in this chapter, it is possible (although little noted) that multiple network addresses could reside on a common network. This is the exception rather than the rule. Similarly, two hosts on different networks should be assigned different network prefixes; however, the hosts can have the same host address. In thinking about this addressing technique, one can consider it in many ways to be similar to the structure of a telephone number. That is, no two people in the same area code can have the same phone number. It is both possible and likely that somewhere the same phone number exists in a different area code.

One can also view Class A, B, and C addresses as having the following general format:

< Network Number, Host Number >

where the combined network number and host number have the form xxxx.xxxx.xxxx.xxxx, with each x representing a decimal value. Probing deeper into IP addressing, one sees that the above format uses dotted decimal notation to reference IP addresses. By the end of this section, the reader will be conversant in the use of this method of IP address notation.

Rationale

During the IP standardization process, it was recognized that a single method of subdivision of the 32-bit address space into network and host portions would be wasteful with respect to the assignment of addresses. For example, assume all addresses are evenly split. This would result in the use of 16 bits for a network number and a similar number of bits for a host number. Without

considering host and network addressing restrictions (discussed later in the section), the use of 16 bits results in a maximum of 65,536 (2^{16}) networks with up to 65,536 hosts per network. Not only would the assignment of a network address to an organization that has only 100 computers result in a waste of 65,436 host addresses that could not be assigned to other organizations, but in addition, there could only be 65,536 networks. This limited number of networks would be clearly insufficient in an era where over 50,000 colleges, universities, high schools, and grade schools are now connected to the Internet via LANs, with each LAN having a distinct network address. Recognizing that the use of IP addresses could literally mushroom beyond their expectations, the designers of IP came up with a methodology whereby the 32-bit IP address space was subdivided into different address classes. The result of the efforts of IP designers was the definition of five address classes, referred to as Class A through Class E.

Overview

Class A addresses were developed for use by organizations with extremely large networks or for assignments to countries. Class B addresses were developed for use by organizations with large networks, while Class C addresses are used by organizations with small networks. Two additional address classes are Class D and Class E. Class D addresses are used for IP multicasting, a technique where a single message is distributed to a group of hosts dispersed across a network. Class E addresses are reserved for experimental use. Unlike Classes A through C that incorporate a two-level IP addressing structure, Classes D and E use a single addressing structure.

Exhibit 4.7 illustrates the structure or format of the five defined IP address classes. In examining the entries in Exhibit 4.7, note that an address identifier of variable length is the prefix to each address class. The address identifier prefix is a single "0" bit for a Class A address, the bits "10" for a Class B address, etc. Because each address identifier is unique, it becomes possible to examine one or more bits in the address identifier portion of the address to determine the address class. Once an address class is identified, the subdivision of the remainder of the address into the network and host address portions can easily be obtained from a table lookup or from predefined data within a program. For example, if a 32-bit address is a Class A address due to the first bit being binary 0, then the next seven bits represent the actual network address, while the remaining 24 bits represent the host address. Similarly, if the first two bits of the 32-bit address have the value "10," then the next 14 bits represent the actual network address, while the trailing 16 bits represent the host address. To obtain an appreciation of the use of each IP address class, a detailed examination of each address class follows, with particular attention to the composition of the network and host portion of each address for Classes A through C, as well as the manner by which all five classes are used.

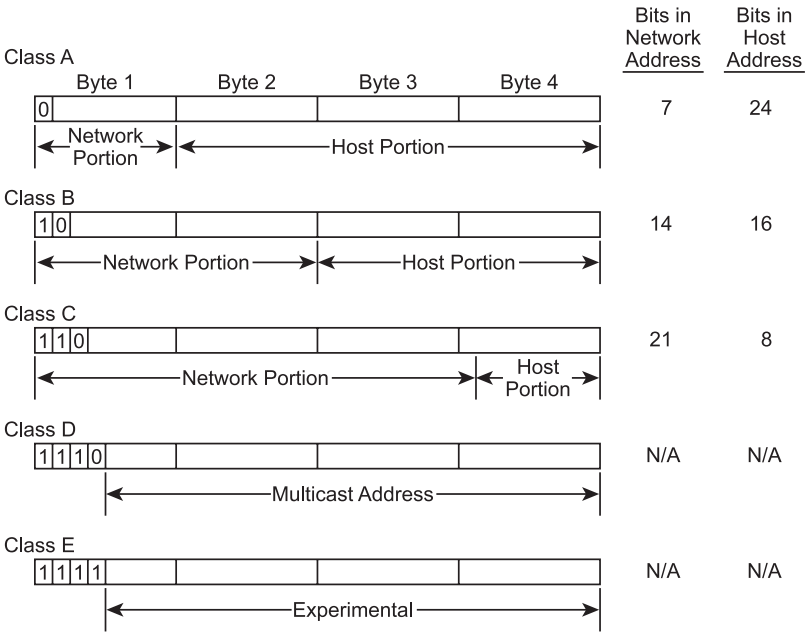


Exhibit 4.7 IP Address Formats

Class A Addresses

As indicated in Exhibit 4.7, a Class A address has the four-byte form of <network-number.host.host.host>, with seven bits used for the actual network address because the first bit position must be set to a value of binary 0 to indicate the address is a Class A address. Because seven bits are available for the network address, one would logically assume 2^7 or 128 Class A networks can be defined. In actuality, networks 0 and 127 are reserved and cannot be used, resulting in Class A addressing supporting 126 networks. Because there are 24 bits used for a host identifier, this means that each network is capable of supporting up to $2^{24} - 2$ or 16,277,214 hosts. The reason 2 is subtracted from the possible number of hosts results from the fact that no host can be assigned a value of all 0s nor a value of all 1s. As noted later in this chapter, a host value of all 1s indicates a broadcast address.

Because only a small number of Class A networks can be defined, they were used up many years ago. Due to the large number of hosts that can be assigned to a Class A network, Class A addresses were primarily assigned to large organizations and countries that have national networks.

Loopback

One Class A network address that warrants attention results from the setting of all seven bits in the network address to 1, representing 127 in decimal. A

network address of 127.x.x.x is reserved as an internal loopback address and cannot be assigned as a unique IP address to a host. Thus, a question one may have is, "why reserve a network address of 127 if it is not usable?" The answer to this question is the fact that one can use a network address of 127.x.x.x as a mechanism to determine if one's computer that loaded TCP/IP protocol stack has an operational stack. An example of the use of a 127-network address is illustrated in the top of Exhibit 4.8, which shows the use of the Ping command to query the device at address 127.1.1.1. Because this is a loopback address, this action tests the protocol stack on the author's computer. Note that in this example, Microsoft's version of Ping uses the IP address 127.1.1.1 as a loopback. If one enters the address 127.0.0.0 as shown in the lower portion of Exhibit 4.8, Microsoft's implementation of the TCP/IP protocol stack treats the IP address as an invalid address.

All TCP/IP protocol stacks should, as a minimum, recognize the IP address 127.0.0.1 as an internal loopback address. Most protocol stacks will also consider a prefix of 127 for a network address with any non-zero host address as a loopback. Thus, one can normally use 127.1.2.3, 127.4.5.6, and any other combination other than 127.0.0.0 as a loopback.

Class B Addresses

Continuing this exploration of IPv4 address classes, a Class B address has the form <network-number.network-number.host.host> for the four bytes in the address. A Class B network address is defined by setting the two high-ordered bits of an IP address to the binary value "10." Because two bits are used to identify the address, this means that the actual Class B network address is 14 bits in width, while the host portion of the address is two bytes or 16 bits in width. Thus, a Class B address is capable of supporting 2^{14} (or 16,384) networks, with each network capable of supporting up to $2^{16} - 2$, (or 65,534) hosts. Due to the manner by which Class B addresses are subdivided into network and host portions, such addresses are normally assigned to relatively large organizations. In addition, through the process of subnetting, which is described later in this section, one Class B address can be provided to multiple organizations, with each organization informed as to the correct subnet mask to use to identify the portion of a Class B address provided for their use.

If familiar with binary, one can easily convert permissible binary values in the first byte of a Class B address into a range of decimal values. For example, because a Class B address commences with binary values 10, the first byte must range between 1000000 and 10111111. One can convert to decimal by noting that the value of each position in a byte is as follows:

128 64 32 16 8 4 2 1

Thus, binary 10000000 is equivalent to decimal 128, while binary 10111111 is equivalent to decimal 191. Thus, the first byte of a Class B address is restricted

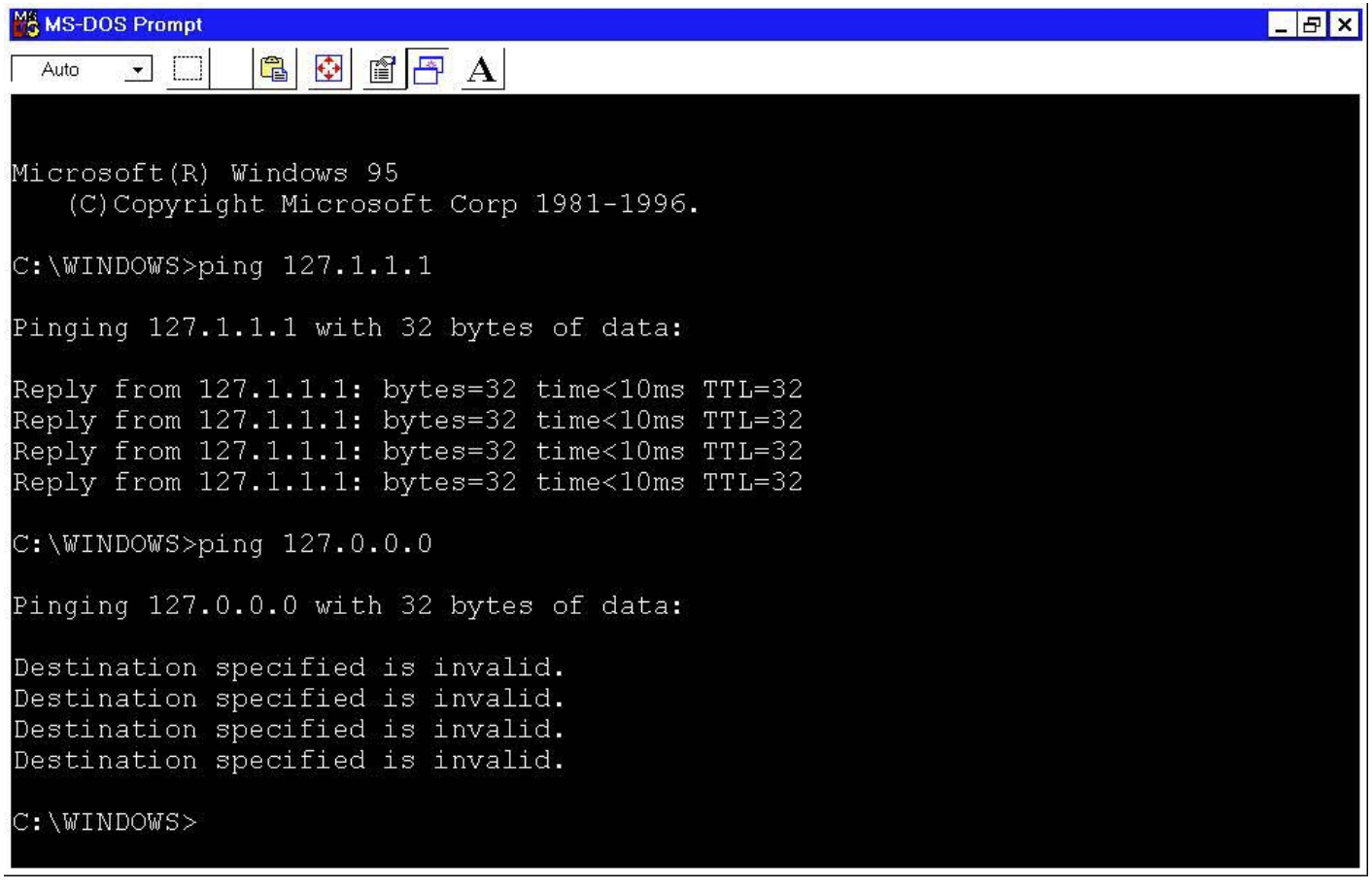

MS-DOS Prompt
Auto
Microsoft(R) Windows 95
(C)Copyright Microsoft Corp 1981-1996.
C:\WINDOWS>ping 127.1.1.1
Pinging 127.1.1.1 with 32 bytes of data:
Reply from 127.1.1.1: bytes=32 time<10ms TTL=32
Reply from 127.1.1.1: bytes=32 time<10ms TTL=32
Reply from 127.1.1.1: bytes=32 time<10ms TTL=32
Reply from 127.1.1.1: bytes=32 time<10ms TTL=32
C:\WINDOWS>ping 127.0.0.0
Pinging 127.0.0.0 with 32 bytes of data:
Destination specified is invalid.
Destination specified is invalid.
Destination specified is invalid.
Destination specified is invalid.
C:\WINDOWS>

Exhibit 4.8 Using an IP Loopback Address with a Ping Application to Verify the Status of the TCP/IP Protocol Stack

to the range 128 to 191, with 0 to 255 permitted in the second byte of the network address.

Class C Addresses

A Class C address is identified by the first three bits in the IP address being set to the binary value of 110. This value denotes the fact that the first three bytes in the 32-bit address identify the network, while the last byte identifies the host on the network. Because the first three bits in a Class C address are set to a value of 110, this means there are 21 bits available for the network address. Thus, a Class C address permits 2^{21} (or 2,097,152) distinct network addresses. Since the host portion of a Class C address is 1 byte in length, the number of hosts per network is limited to $2^8 - 2$ (or 254).

Due to the subdivision of network and host portions of Class C addresses, they are primarily assigned for use by organizations with relatively small networks, such as a single LAN that requires a connection to the Internet. Because it is common for organizations to have multiple LANs, it is also quite common for multiple Class C addresses to be assigned to organizations that require more than 254 host addresses, but are not large enough to justify a Class B address. It is also common for an organization with multiple LANs located within close proximity to one another to share one Class C address through subnetting, a topic covered later in this chapter.

Similar to the manner in the decimal range of Class B addresses was computed, one can compute the range of permitted Class C addresses. That is, because the first three bits in the first byte are set to a value of 110, the binary range of values are 11000000 to 11011111, representing decimal 192 through 223. The second and third bytes in a Class C address range in value from 0 to 255, while the last byte, which represents the host address, ranges in value from 1 to 254, because host values of 0 and 255 are not permitted.

Class D Addresses

Class D IP addresses represent a special type of address referred to as a multicast address. A muticast address is assigned to a group of network devices and allows a single copy of a datagram to be transmitted to a specific group. The members of the group are then able to receive a common sequence of datagrams instead of having individual series of datagrams transmitted to each member on an individual basis, in effect conserving network bandwidth.

A Class D address is identified by the assignment of the binary value 1110 to the first four bits of the address. The remaining 28 bits are then used to define a unique multicast address. Because a Class D address always has the prefix 1110, its first byte varies from 11100000 to 11101111, resulting in the address range 224 through 239. Thus, the multicast address range becomes 224.0.0.0 through 239.255.255.255, with the use of a Class D address enabling approximately 268 million multicast sessions to simultaneously occur throughout the world.

To obtain an appreciation for the manner by which Class D addressing conserves bandwidth, consider a digitized audio or video presentation routed from the Internet onto a private network for which users working at 15 hosts on the private network wish to receive the presentation. Without a multicast transmission capability, 15 separate data streams, each containing a repetition of the audio or video presentation, would be transmitted through the Internet onto the private network, with only the destination address in each datagram in one stream differing from the datagram in a different stream. Here, 14 data streams are unnecessary and only function to clog the Internet as well as the private network. In comparison, through the use of multicasting, the 15 users requiring the presentation would join the multicast group, permitting one data stream to be routed through the Internet onto the private network.

Common examples of the use of multicast include access to many news organization video feeds that result in a 2 × 2-inch television on a computer monitor. With frame refresh rates of 15 or more frames per second, a server of Unicast transmissions would consume a relatively large amount of bandwidth. Thus, the ability to eliminate multiple data streams via multicast transmission can prevent networks from being saturated. In addition, this capability reduces the number of datagrams that routers must route. This minimizes the necessity of routers that discard packets when they become saturated.

Class E Addresses

The fifth address class defined for IPv4 is Class E. A Class E address is defined by the setting of the first four bits in the 32-bit IP address to the binary value of 1111. Thus, a Class E address has a first byte value between 11110000 and 11111111, or between 240 and 255 decimal.

Class E addresses are currently reserved for experimental usage. Because there are 28 bits in a Class E address that can be used to define unique addresses, this means there are approximately 268.4 million available Class E addresses.

One common method used to denote Class A through E addresses is by examining the decimal value of the fist byte of the 32-bit IPv4 address. To facilitate this examination, Exhibit 4.9 summarizes the range of decimal values for the first byte of each address class.

Exhibit 4.9 IPv4 Address Class First Byte Values

Address Class	*First Byte Address Range*
Class A	1 to 126
Class B	128 to 191
Class C	192 to 223
Class D	224 to 239
Class E	240 to 255

Dotted Decimal Notation

Only a brief examination of how to convert the binary value of a byte into decimal has been given, with no discussion of the rationale for the use of decimal numbers in IP addresses. Thus, the rationale is presented here.

Because humans do not like to work with strings of 32-bit binary addresses, the developers of IP looked for a technique that would make it easier to specify IPv4 addresses. The resulting technique is referred to as "dotted decimal notation," in recognition of the fact that a 32-bit IP number can be subdivided into four eight-bit bytes. Because of this, it is possible to specify a 32-bit IPv4 address via the use of four decimal numbers in the range 0 through 255, with each number separated from another number by a decimal point.

To review the formation of a dotted decimal number, first focus on the decimal relationship of the bit positions in a byte. Exhibit 4.10 indicates the decimal values of the bit positions within an eight-bit byte. Note that the decimal value of each bit position corresponds to 2^n, where "n" is the bit position in the byte. Using the decimal values of the bit positions shown in Exhibit 4.10, assume one wants to convert the following 32-bit binary address into dotted decimal notation:

01010100110011101111000100111101

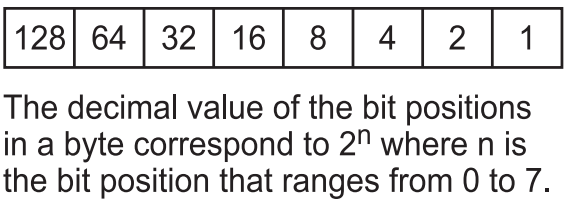


Exhibit 4.10 Decimal Values of Bit Positions in a Byte

The first eight bits that correspond to the first byte in an IP address have the binary value 01010100. Then, the value of that byte expressed as a decimal number becomes 64 + 16 + 4, or 84. Next, the second bit in the binary string has the binary value of 11001110. From Exhibit 4.10, the decimal value of the second byte is 128 + 64 + 8 + 4 + 2, or 206. Similarly, the third byte, whose binary value is 11110001, has the decimal value 128 + 64 + 32 + 16 + 1, or 241. The last byte whose bit value is 00111101 would have the decimal value 32 + 16 + 8 + 4 + 1, or 61. Based on the preceding, one would enter the 32-bit address in dotted decimal notation as 84.206.241.61, which is certainly easier to work with than a 32-bit string.

Basic Workstation Configuration

The use of dotted decimal notation can be appreciated when examining the configuration of a workstation. If using Microsoft Windows 95 or Windows 98,

one would go to Start> Control Panel> Network and double-click on the TCP/IP entry in the configuration tab to assign an applicable series of dotted decimal values to configure a host on an IP network.

To correctly configure a host on a TCP/IP network requires the entry of three dotted decimal addresses and a subnet mask, the latter also specified as a dotted decimal number. The three addresses one must specify include the IP address of the host being configured, the IP address of a gateway, and the IP address of a domain name server.

The term "gateway" dates from the early days of ARPAnet when a device that routed datagrams between networks was referred to by that name. Today, this device is referred to as a router; however, in the wonderful world of TCP/IP configuration, the term "gateway" is still used. The second new device is the DNS that resolves (a fancy name for translates) host names into IP addresses, and its operation will be described in more detail later in this book (Chapter 6). At the present time, simply note that the DNS allows one to enter addresses into Web browsers, such as www.whitehouse.gov, and allows the TCP/IP protocol stack to perform the translation into an applicable IP address. All routing in an IP network occurs via an examination of IP addresses.

Exhibit 4.11 illustrates the setting of the IP address tab in the TCP/IP Properties dialog box on the author's personal computer. Note that the button labeled "Specify an IP address" is shown selected, which indicates to the Windows operating system that a fixed IP address will be assigned to the computer. In Exhibit 4.11, that address is 198.78.46.8, which, if one converts 198 into binary rather than glancing at Exhibit 4.9, one will note a value of 11000000. Because the first three bits are set to binary 110, this denotes a Class C address. If one does not like working with binary, one could then use Exhibit 4.9 to determine that the setting of the first byte to 198 is indeed a Class C address.

Although the subnet mask will be discussed shortly, at the present time one can note here that its setting "extends" the network portion of an address internally within an organization. That is, the set bits in a subnet mask indicate the new length of the network portion of the address. Examining the subnet mask shown in Exhibit 4.11 and remembering that a value of 255 represents the setting of all bits in a byte to 1, this indicates that the network portion of the address is 24 bits long. Because a Class C address uses three bytes for the network address and one byte for the host address, this also means that a subnet mask of 255.255.255.0 for a Class C address indicates that the network is NOT subnetted.

By clicingk on the tab labeled "gateway," one can view the manner by which one can add and remove the IP addresses of routers. Exhibit 4.12 illustrates the TCP/IP Properties dialog box with its gateway tab selected. In this example, the IP address 198.78.46.1 was entered to denote the address of the router that will route datagrams with an IP network address other than 198.78.46.0 off the network.

The third IP address used for the configuration of a TCP/IP protocol stack is the address of a DNS server that supports an organization's network. One can view the DNS configuration screen by clicking on the tab with that label.

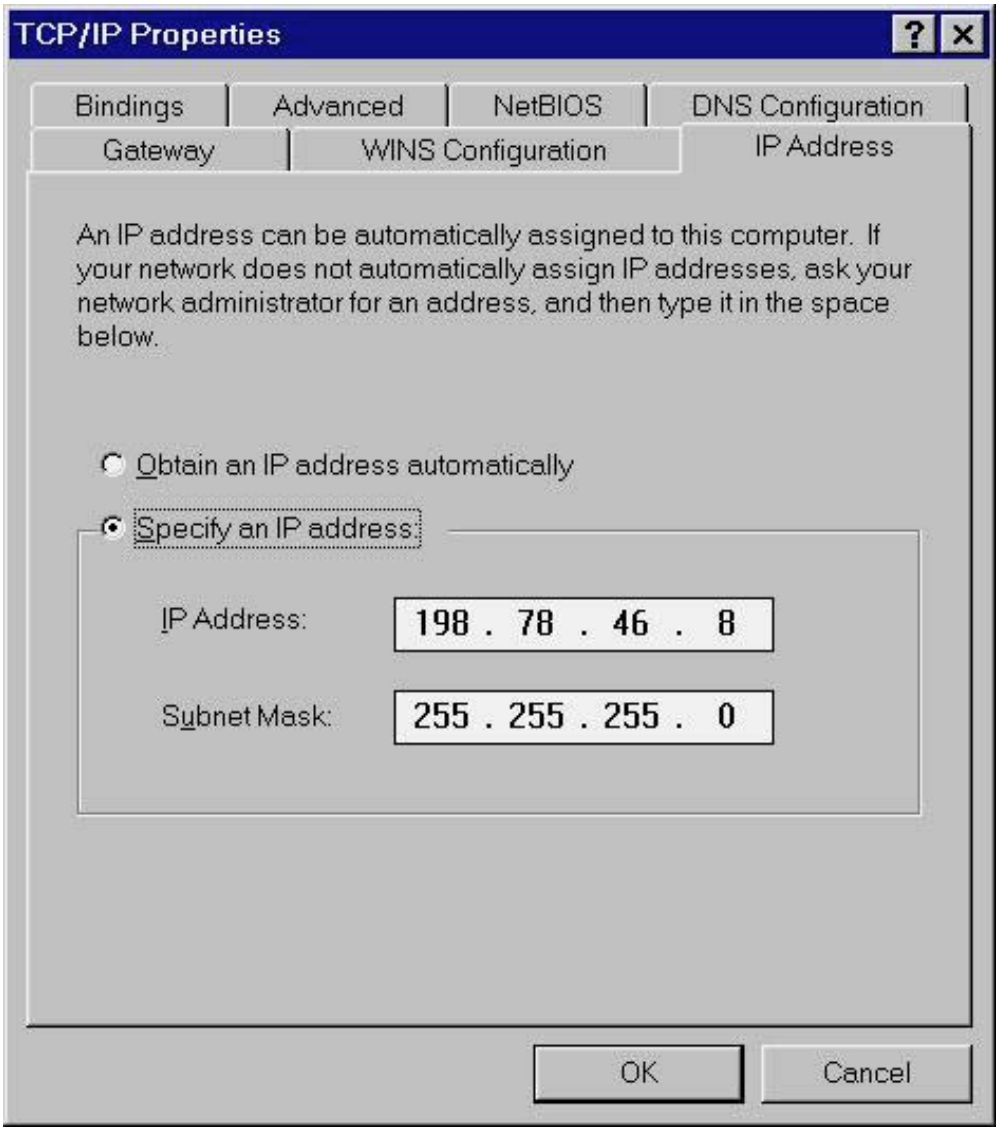

TCP/IP Properties
Bindings
Advanced
NetBIOS
DNS Configuration
Gateway
WINS Configuration
IP Address
An IP address can be automatically assigned to this computer. If your network does not automatically assign IP addresses, ask your network administrator for an address, and then type it in the space below.
Obtain an IP address automatically
Specify an IP address:
IP Address:
198 . 78 . 46 . 8
Subnet Mask:
255 . 255 . 255 . 0
OK
Cancel

Exhibit 4.11 Setting the IP Address and Subnet Mask

Exhibit 4.13 illustrates the TCP/IP Properties dialog box with its DNS configuration tab selected. Note that the radio button associated with Enable DNS is shown selected, and a host name of "gil" was entered for this computer which is part of the domain fed.gov. Thus, the complete host name of this computer is gil.fed.gov. Note that one does not have to specify either host or domain. Doing so results in the IP address previously assigned to this computer, along with the host name entered in a record in the DNS server. This would then allow someone to access this computer by entering gil.fed.gov instead of the IP address of 198.78.46.8. If no one accesses the computer, one could safely omit the host and domain entries. If the computer is a popularly

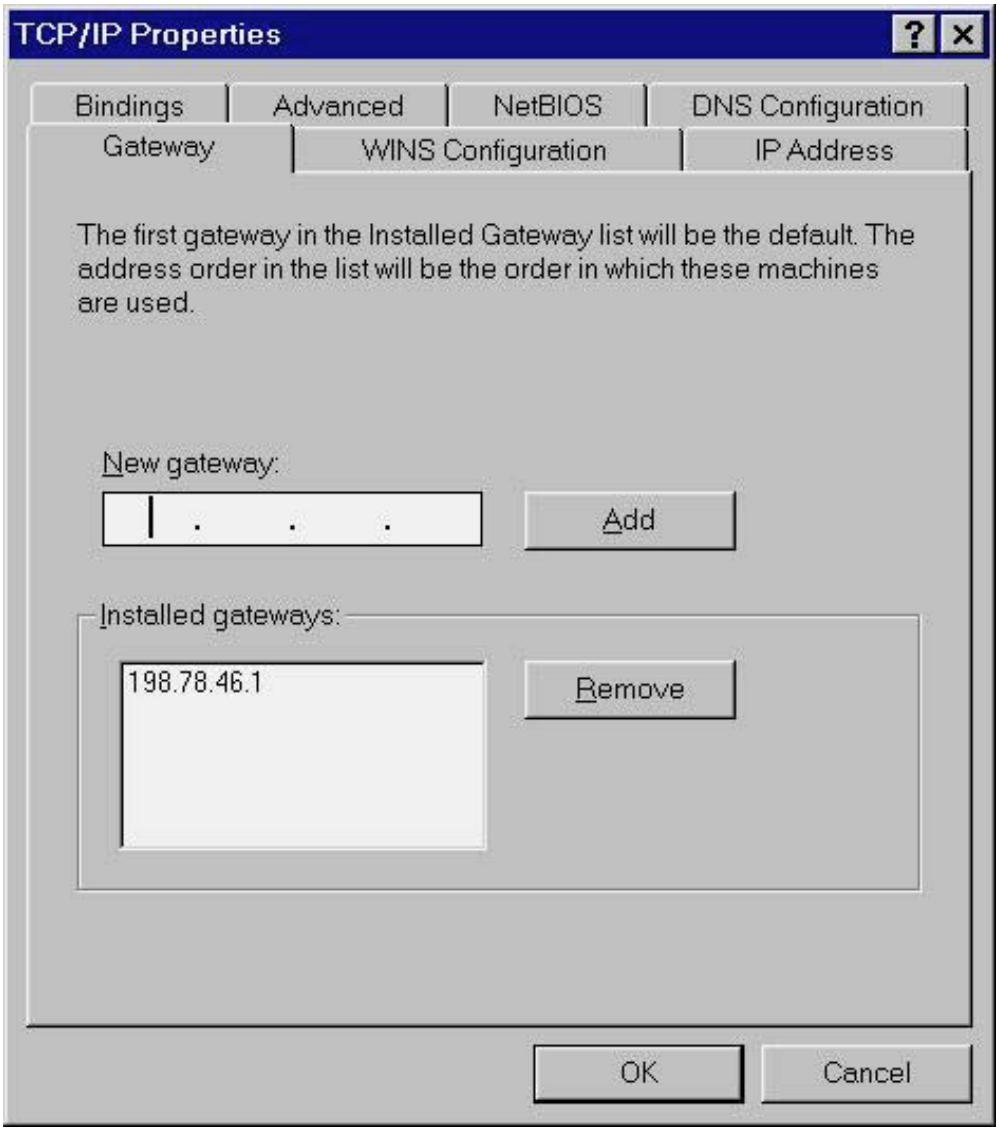


Exhibit 4.12 Configuring the Gateway Address under Windows 95/98

used server, one would want to include the host name because it would be easier to remember than a sequence of dotted decimal numbers.

The combination of host and domain is commonly referred to as a fully qualified domain name (FQDN). An FQDN means that the name is unique. In comparison, the host portion of the name (gil) could exist on many domains. Similarly, many computers could have a common domain name (fed.gov).

Returning to Exhibit 4.13, note that one can specify up to four DNS server addresses. In addition, one can specify one or more domain suffix search orders where common domain suffixes include gov (government), com (commercial), edu (educational), mil (military), and org (nonprofit organization).

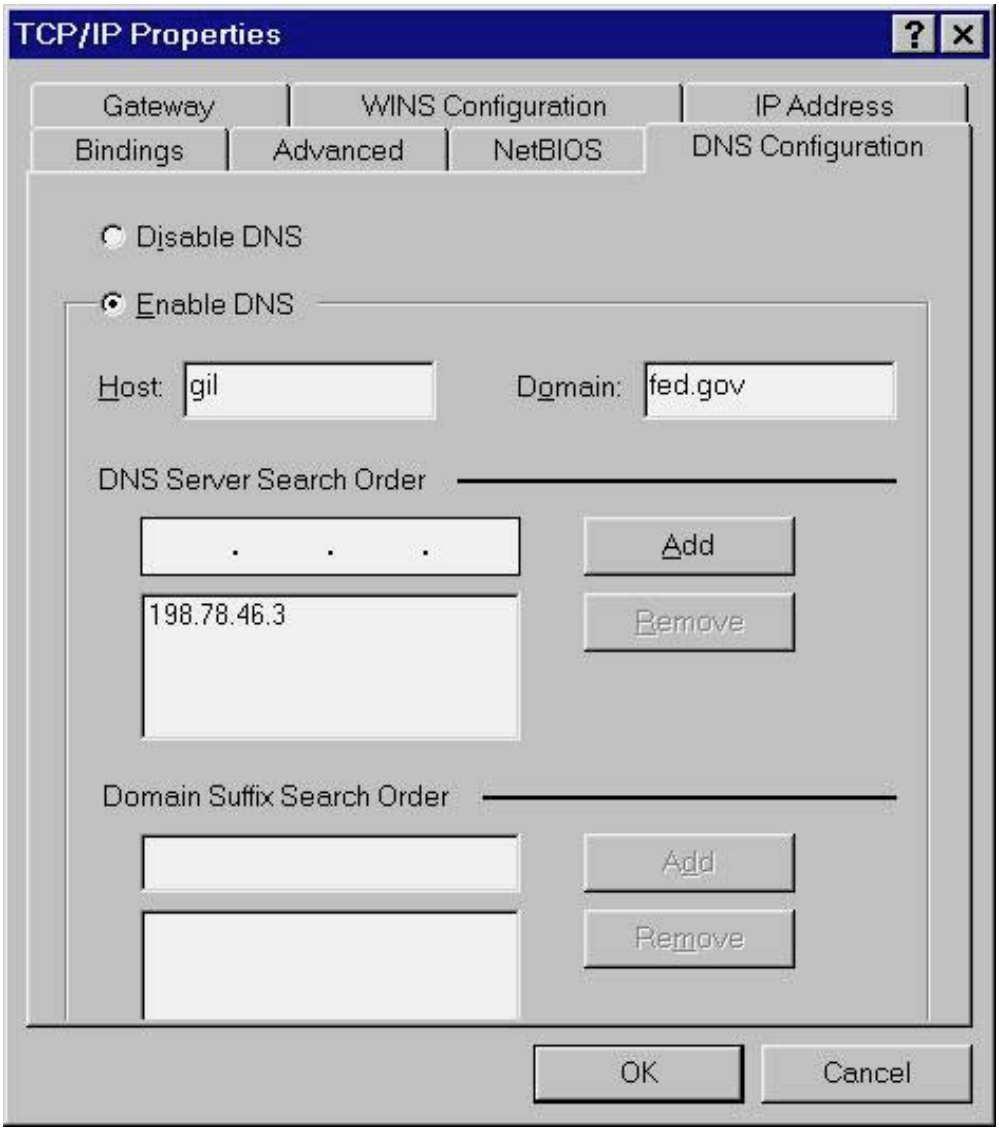


Exhibit 4.13 Specifying the Address of the DNS Server and the Fully Qualified Name of the Host at the DNS Tab

Reserved Addresses

It was previously noted that the address block 127.0.0.0 through 127.255.255.255 is used for loopback purposes and can thus be considered to represent a block of reserved addresses. When considering IPv4 addressing, there are three additional blocks of reserved addresses that warrant attention. Those address blocks are defined in RFC 1918, entitled *Address Allocation for Private Internet,* and are summarized in Exhibit 4.14.

Exhibit 4.14 Reserved IP Addresses for Private Internet Use (RFC 1918)

Address Blocks
10.0.0.0–10.255.255.255
172.16.0.0–172.31.255.255
192.168.0.0–192.168.255.255

The original intention of RFC 1918 addresses was to define blocks of IP addresses organizations could use on private networks that would be recognized as such. As the use of the Internet grew, the ability to obtain IP addresses became more difficult because existing network addresses were assigned to different organizations. This resulted in a second role for RFC 1918 addresses under a process referred to as Network Address Translation (NAT). Under NAT, internal RFC 1918 addresses can be dynamically translated to public IP addresses while reducing the number of public addresses that need to be used. For example, consider an organization with 500 stations that only has one Class C address. One possibility is to use RFC 1918 addresses behind a router connected to the Internet, with the router translating RFC 1918 addresses dynamically into available Class C addresses. Although no more that 254 RFC 1918 addresses could be translated into valid distinct Class C addresses at any point in time, it is also possible to use TCP and UDP port numbers to extend the translation process so each RFC 1918 address can be simultaneously used and translated. To do so, a router would translate each RFC 1918 address into a Class C address using a different port number, permitting thousands of translations for each Class C address.

Another device that can provide address translation is a proxy firewall. In addition to translating addresses, a proxy firewall also hides internal addresses from the Internet community. This address hiding provides a degree of security because any hacker that attempts to attack a host on a network where a proxy firewall operates must first attack the firewall.

Two additional items to note about RFC 1918 addresses are that they cannot be used directly on the Internet, and they are a favorite source address used by hackers. The reason RFC 1918 addresses cannot be directly used on the Internet results from the fact that if one company does so, a second could also do so, resulting in addressing conflicts and the unreliable delivery of information. Thus, as discussed, RFC 1918 addresses are translated into Class A, B, or C addresses when a private network using such addresses is connected to the Internet. Concerning hacker use, because source IP addresses are not checked by routers, it is quite common for an RFC 1918 address to be used as the source address by a hacker, making it difficult — if not impossible — to locate the hacker.

Because it is quite common for hackers to use an RFC 1918 address as their address in configuring a TCP/IP protocol stack, it is also quite common

to create a router access list that filters datagrams that have an RFC 1918 address. When network security is discussed in Chapter 9, also included will be applicable access list statements to send datagrams with RFC 1918 source addresses to the great bit bucket in the sky.

Subnetting

One of the problems associated with the use of IP addresses is the fact that even with the use of classes, their use can be inefficient. For example, consider the use of a Class A network address. Although one can have up to 16,277,214 hosts per Class A network, one can only have 127 such networks. Thus, the assignment of a Class A network address to a large organization with 100,000 workstations would waste over 16 million IP addresses. Similarly, because a single LAN is incapable of supporting 100,000 workstations, one might consider asking for multiple network addresses, which would further waste a precious resource referred to as IPv4 addresses. Another problem associated with using more network addresses than required is the fact that routers must note those addresses. This means that the routers in a network, which could be the Internet or a private TCP/IP network, would have more entries in its routing tables. This, in turn, results in routers requiring a longer time to check the destination address in a datagram against entries in each router's routing table. The solution to the problems of wasted IP address space and unnecessary routing table entries is provided through the process of subnetting.

Overview

Subnetting was standardized in RFC 950 in 1985. This RFC defines a procedure to subnet or divide a single Class A, B, or C network into two or more subnets. Through the process of subnetting, the two-level hierarchy of Class A, B, and C networks previously illustrated in Exhibit 4.6 is converted into a three-level hierarchy. Exhibit 4.15 provides a comparison between the two-level hierarchies initially defined for Class A, B, and C networks and the three-level subnet hierarchy. In examining the lower portion of Exhibit 4.15, note that to convert the two-level hierarchy into a three-level hierarchy, the extension of the network address occurs by taking away a portion of the host address portion of an IPv4 address.

Subnetting Example

As previously noted, any of the IPv4 A through C address classes can be subnetted. To illustrate the subnet process, as well as obtain an appreciation for how subnetting facilitates the use of IPv4 address space, one can examine the process by understanding the concept of masking and the use of the subnet mask, both of which are essential to the extension of the network portion of an IP address beyond its predefined location.

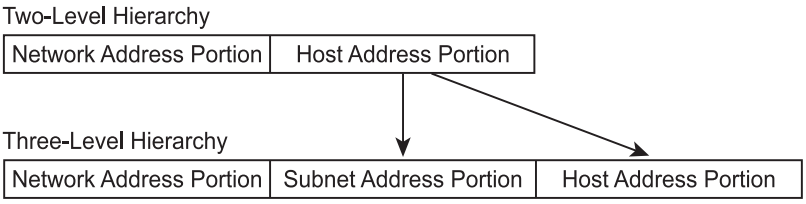


Exhibit 4.15 Comparing the Three-Level Subnet Hierarchy to the Two-Level Network Class Hierarchy

To illustrate the concept of subnetting, assume on organization has the need to install five LANs within a building, with each network supporting between 10 and 15 workstations and servers. Further assume that the organization was previously assigned the IP Class C network address 198.78.46.0. Although the organization could apply for four additional Class C addresses, doing so would waste precious IPv4 address space, because each Class C address supports a maximum of 254 interfaces. In addition, if one anticipates connecting the organization's private networks to the Internet, the use of four additional Class C network addresses would be required in a number of routers in the Internet as well as the organization's internal routers.

Instead of asking for four additional Class C addresses, one can use subnetting by dividing the host portion of the 198.78.46.0 IPv4 address into a subnet number and a host number. Because one needs to support five networks, one must use a minimum of three bits from the host portion of the IP address as the subnet number. The reason a minimum of three bits from the host portion of the address must be used is due to the fact that the number of subnets one can obtain is 2^n, where n is the number of bits. When n = 2, this yields four subnets, which is too few. When n = 3, one obtains eight subnets, which provides enough subnets for this example.

Because a Class C address uses 24 for the network portion and eight bits for the host portion, the use of a three-bit subnet extends the network address such that it becomes 27 bits in length. This also means that a maximum of five bits (8 – 3) can be used for the host portion of the address.

Exhibit 4.16 illustrates the creation of the three-level addressing scheme just described. Note that the three-bit subnet permits eight subnets (000 through 111).

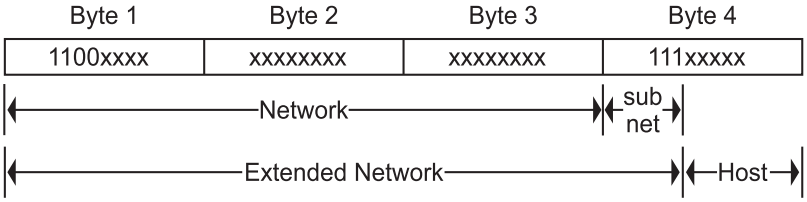


Exhibit 4.16 Creating a Class C Three-Level Addressing Scheme

To the outside world, the network portion of the address remains the same. This means that the route from the Internet to any subnet of a given IP network address remains the same. This also means that routers within an organization must be able to differentiate between different subnets; however, routers outside the organization do not consider subnets.

To illustrate the creation of five subnets, assume one wants to commence subnet numbering at 0 and continue in sequence through subnet 4. Exhibit 4.17 illustrates the creation of five subnets from the 198.78.46.0 network address. Note that the top entry in Exhibit 4.17, which is labeled "Base Network," represents the Class C network address with a host address byte field set to all zeroes. Because it was previously determined that three bits from the host address portion of the network would be required to function as a subnet identifier, the network address is shown extended into the host byte by three portions.

Exhibit 4.17 Creating Extended Network Prefixes via Subnetting

```
Base Network:1100110.01010000.00101110.00000000 = 198.78.46.0
    Subnet #0:1100110.01010000.00101110.00000000 = 198.78.46.0
    Subnet #1:1100110.01010000.00101110.00100000 = 198.78.46.32
    Subnet #2:1100110.01010000.00101110.01000000 = 198.78.46.64
    Subnet #3:1100110.01010000.00101110.01100000 = 198.78.46.96
    Subnet #4:1100110.01010000.00101110.10000000 = 198.78.46.128
```

Host Restrictions

In examining the subnets formed in Exhibit 4.17, it would appear that the hosts on the first subnet can range from 0 through 31, while the hosts on the second subnet can range in value from 33 through 63, etc. In actuality, this is not correct because there are several restrictions concerning host addresses on subnets. First, one cannot use a base subnet address of all zeroes nor all ones. Thus, for subnet 0 in Exhibit 4.17, valid addresses would range from 1 to 30. Similarly for subnet 1, valid addresses would range from 33 to 62. Thus, subnetted host address restrictions are the same as for a regular IP non-subnetted network.

Another host address restriction that requires consideration is the fact that for all classes, one must have the ability to place some hosts on each subnet. Thus, as a minimum, the last two bit positions into the fourth byte of Class A, B, and C addresses cannot be used in a subnet. Exhibit 4.18 illustrates the number of bits that are available for subnetting for Class A, B, and C network addresses.

The Zero Subnet

Another item concerning subnetting that warrants attention is the fact that at one time, the zero subnet was considered anathema by the Internet community,

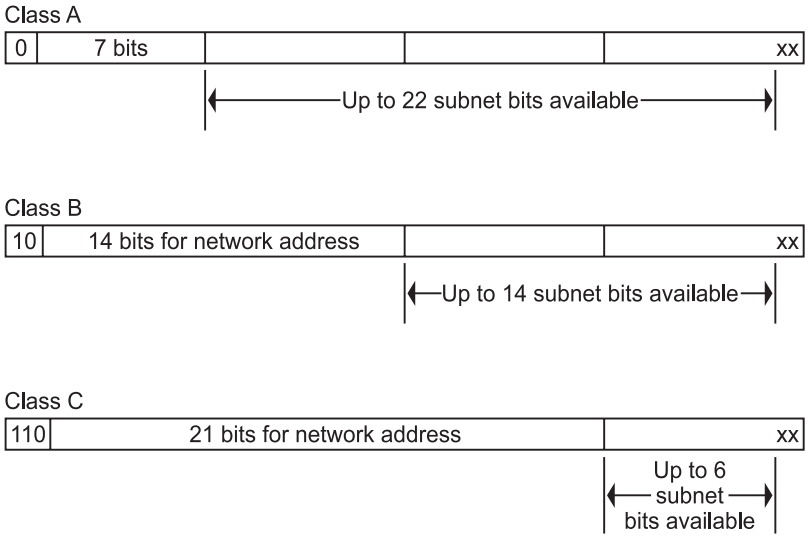


Exhibit 4.18 Available Bit Positions for Subnet Formation

and its use was and to a degree still is discouraged. While this viewpoint has somewhat fallen from favor, it is important to note that some devices will not support the use of subnet zero and will not allow one to configure their interface address as being on a zero subnet. The reason for this restriction results because confusion can arise between a network and a subnet that have the same address. For example, assume network address 129.110.0.0 is subnetted as 255.255.255.9. This would result in subnet zero being written as 129.110.0.0, which is the same as the network address.

When configuring TCP/IP devices, it is important to note that some devices that support a zero subnet must be explicitly configured to do so. For example, the most popular manufacturer of routers is Cisco Systems. Although all Cisco routers support the use of subnet zero, one must use the router command ip subnet-zero to configure a Cisco router to do so. If one attempts to configure a subnet zero, one will receive an "inconsistant network mask" error message.

Internal Versus External Subnet Viewing

Returning to the subnetting example in which five subnets were created from one Class C network address, one can easily understand why subnetting saves router table entries. This is illustrated in Exhibit 4.19, which depicts an internal intranet view of the use of subnets versus a view from the Internet for the prior example. In examining Exhibit 4.19, note that all five subnets appear as the IP network address 198.78.46.0 to routers on the Internet. This means that each router must have knowledge of one IP network address. At the router connected to the Internet, that device becomes responsible for examining each inbound datagram and determining the appropriate subnet where the datagram should be routed. To do so, this router uses a subnet mask whose

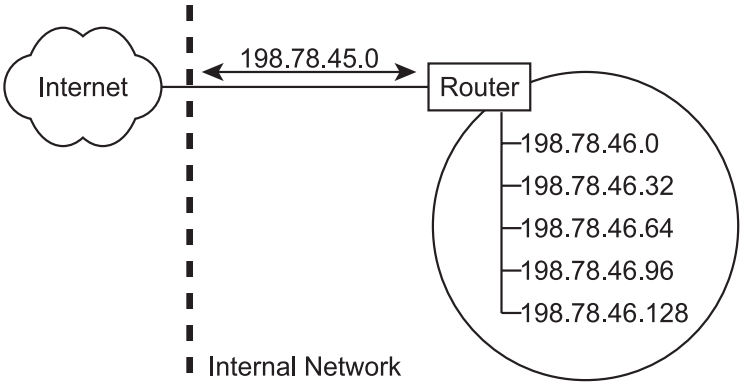


Exhibit 4.19 Internet versus Internal Network View of Subnets

composition and use are discussed below. Prior to doing so, a few points concerning the use of the base network address of 198.78.46.0 are in order. First, to each router the destination address in each datagram appears as a 32-bit sequence. Thus, there is no knowledge of dotted decimal numbers except for the configuration of devices because routing occurs by the examination of the network portion of the address in each datagram. Second, each router begins its address examination by first focusing attention on the first bit in the destination address to determine if it is a Class A address. If the first bit position is set to a binary "0," the router knows it is a Class A address, as well as knows that the first byte in the 32-bit destination address represents the network address. Similarly, if the first bit in the destination address is not a binary "0," the router examines the second bit to determine if the address is a Class B address, etc. Thus, a router can easily determine the address class of the destination address in a datagram that then indicates the length of the network portion of the address. The router can then use this information to search its routing table entries to determine the appropriate port to output the datagram, all without having to consider whether or not the address represents a subnetted address.

Thus far, this chapter has discussed how to create a subnet and extend the network portion of an IPv4 address, but has not addressed the manner by which a router at the edge of the Internet knows how to route datagrams to their appropriate subnet. In addition, there is the question of how a station on an internal network can recognize subnet addressing. For example, if an IP datagram arrives at an organizational router with the destination address 198.78.46.38, how does the router know to place the datagram on subnet 1? The answer to these questions is the use of a subnet mask.

Using the Subnet Mask

The subnet mask provides a mechanism that enables devices to determine the separation of an IPv4 address into its three-level hierarchy of network,

subnet, and host addresses. To accomplish this task, the subnet mask consists of a sequence of set to "1" bits that denotes the length of the network and subnet portions of the IPv4 network address associated with a network. That is, the subnet mask indicates the internal extended network address.

To illustrate the use of the subnet mask, again assume the network address to be 198.78.46.0. Further assume that one wants to create a subnet mask that can be used by a router or workstation to note that the range of permissible subnets is 0 to 7. Because this requires the use of three bits, the subnet mask becomes:

11111111.11111111.11111111.11100000

Similar to the manner by which IP addresses can be expressed more efficiently through the use of dotted decimal notation, one can also express subnet masks using that notation. Because each byte of all set bits has a decimal value of 255, the dotted decimal notation for the first three bytes of the subnet mask is 255.255.255. Because the first three bits of the fourth byte are set, its decimal value is 128 + 64 + 32, or 224. Thus, the dotted decimal specification for the subnet mask becomes:

255.255.255.244

Because a device can easily determine the address class of the destination address in a datagram, the subnet mask then informs the device of which bits in the address represent the subnet and indirectly which bits represent the host address on the subnet. To illustrate how this is accomplished, assume a datagram has arrived at a router with the destination IP address 198.78.46.97, and that the subnet mask was previously set to 255.255.255.224. The relationship between the IP address and the subnet mask would then appear as indicated in Exhibit 4.20.

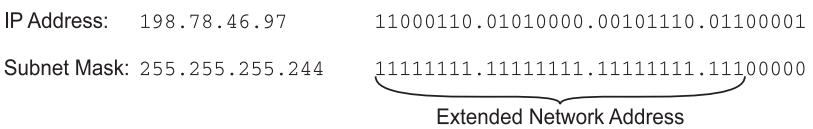


Exhibit 4.20 Examining the Relationship between an IP Address and a Subnet Mask

Because the first two bits in the destination address are set to 11, this indicates the address is a Class C address. The TCP/IP protocol stack knows that a Class C address consists of three bytes used for the network address, and one byte used for the host address. Thus, this means that the subnet must be 27 – 24, or three bits in length. This fact tells the router or workstation that bits 25 through 27, which are set to a value of 011 in the IP address,

identify the subnet as subnet 3. Because the last five bits in the subnet mask are set to zero, this indicates that those bit positions in the IP address identify the host on subnet 3. Because the setting of those five bits have the value 00001, this means that the IP address of 198.78.46.97 references host 1 on subnet 3 on the IPv4 network 198.78.46.0.

To assist readers who need to work with subnets, Exhibit 4.21 provides a reference to the number of subnets that can be created for Class B and Class C networks, their subnet mask, the number of hosts per network, and the total number of hosts supported by a particular subnet mask. In examining the entries in Exhibit 4.21, one notes that the total number of hosts can vary considerably, based on the use of different length subnet extensions. Thus, one should carefully consider the effect of a potential subnetting process prior to actually performing the process.

Exhibit 4.21 Class B and Class C Subnet Mask Reference

Number of Subnet bits	*Subnet Mask*	*Number of Subnetworks*	*Hosts/ Subnet*	*Total Number of Hosts*
Class B				
1	—	—	—	—
2	255.255.192.0	2	16382	32764
3	255.255.224.0	6	8190	49140
4	255.255.240.0	14	4094	57316
5	255.255.248.0	30	2046	61380
6	255.255.252.0	62	1022	63364
7	255.255.254.0	126	510	64260
8	255.255.255.0	254	254	64516
9	255.255.255.128	510	126	64260
10	255.255.255.192	1022	62	63364
11	255.255.255.224	2046	30	61380
12	255.255.255.240	4094	14	57316
13	255.255.255.248	8190	6	49140
14	255.255.255.252	16382	2	32764
15	—	—	—	—
16	—	—	—	—
Class C				
1	—	—	—	—
2	255.255.255.192	2	62	124
3	255.255.255.224	6	30	180
4	255.255.255.240	14	14	196
5	255.255.255.248	30	6	170
6	255.255.255.252	62	2	124
7	—	—	—	—
8	—	—	—	—

Multiple Interface Addresses

One of the lesser-known aspects of IP addressing is the fact that it is possible to assign multiple logical network addresses to one physical network. Prior to examining how this occurs, one will probably want to understand the rationale for doing this. Thus, let us assume an organization originally operated a 10BASE-5 network with 100 users and wants to construct a distributed network within a building that will consist of 350 workstations and server. Further assume that the organization's previously installed 10BASE-5 coaxial-based backbone will be used by adding 10BASE-T hubs to the backbone, with a single router providing a connection to the Internet.

If the organization previously obtained a Class C address when it operated a 10BASE-5 network, adding 250 stations means that a second router interface and two networks would be required because each Class C address supports a maximum of 254 hosts.

TCP/IP supports the ability to assign multiple network addresses to a common interface. In fact, TCP/IP also supports the assignment of multiple subnet numbers to a common interface. This can only be accomplished through the use of a router. Exhibit 4.22 illustrates an example in which three network addresses were assigned to one interface. For low volumes of network traffic, this represents an interesting technique to reduce the number of costly router interfaces required.

As indicated in Exhibit 4.22, the router connection to the coaxial cable would result in the assignment of two IP addresses to its interface — one for each network. In this example, the addresses 205.131.175.1 and 205.131.176.1 were assigned to the router interface. Conversations between devices on the

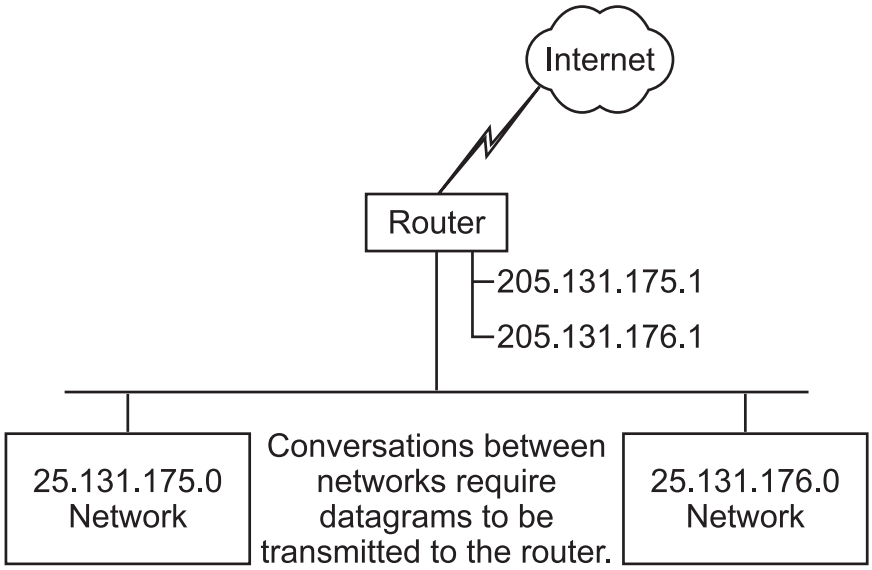


Exhibit 4.22 Assigning Multiple Network Addresses to a Common Router Interface

205.131.175.0 network and the 205.131.176.0 network would require datagrams to be forwarded to the router. Thus, each station of each network would be configured with the "gateway" IP address that represents an applicable assigned router IP interface address.

Address Resolution

The TCP/IP protocol suite begins at the network layer, with an addressing scheme that identifies a network address and a host address for Class A, B, and C addresses. This addressing scheme actually evolved from an ARPAnet scheme that only required hosts to be identified, because that network began as a mechanism to interconnect hosts via serial communications lines. At the same time ARPAnet was being developed, work progressed separately at the Xerox Palo Alto Research Center (PARC) on Ethernet, a technology in which multiple stations were originally connected to a coaxial cable.

Ethernet used a 48-bit address to identify each station on the network. As ARPAnet evolved as a mechanism to interconnect multiple hosts on geographically separated networks, IPv4 addressing evolved into a mechanism to distinguish the network and the host. Unfortunately, the addressing used by the TCP/IP protocol suite bore no relationship to the MAC address used first by Ethernet and later by Token Ring.

Ethernet and Token Ring Frame Formats

Exhibit 4.23 illustrates the frame formats for Ethernet and Token Ring. Note that the IEEE standardized both types of LANs and uses six-byte (48-bit) source and destination addresses. The IEEE assigns blocks of addresses six hex

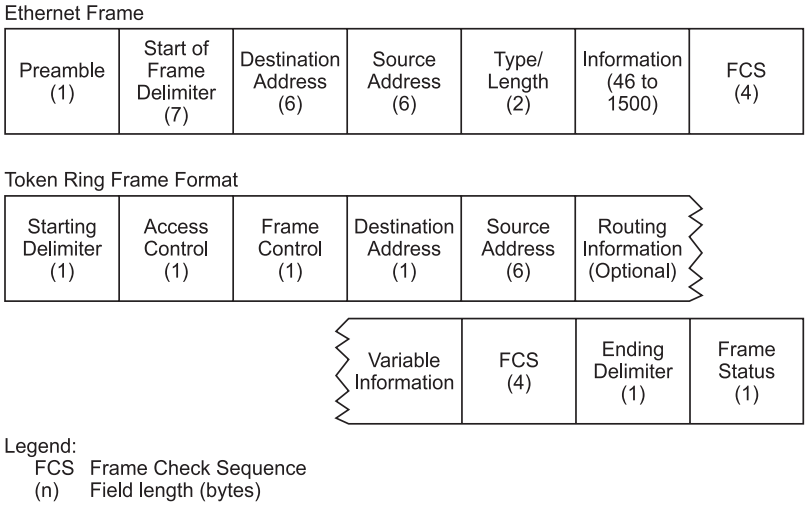


Exhibit 4.23 Ethernet and Token Ring Frame Formats

characters in length to vendors. Those six hex characters represent the first 24 bits of the 48-bit field used to uniquely identify a network adapter card. The vendor then encodes the remaining 24 bits or six hex character positions to identify the adapter card manufactured by the vendor. Thus, each Ethernet and Token Ring adapter has a unique hardware burnt-in identifier that denotes both the manufacturer and the adapter number produced by the manufacturer.

LAN Delivery

When an IP datagram arrives at a LAN, it contains a 32-bit destination address. To deliver the datagram to its destination, the router must create a LAN frame with an appropriate MAC destination address. Thus, the router needs a mechanism to resolve or convert the IP address into the MAC address of the workstation configured with the destination IP address. In the opposite direction, a workstation may need to transmit an IP datagram to another workstation. In this situation, the workstation must be able to convert a MAC address into an IP address. Both of these address translation requirements are handled by protocols specifically developed to provide an address resolution capability. One protocol, referred to as the Address Resolution Protocol (ARP), translates an IP address into a hardware address. A second protocol, referred to as the Reverse Address Resolution Protocol (RARP), performs a reverse translation process, converting a hardware layer address into an IP address.

Address Resolution Operation

The address resolution operation begins when a device needs to transmit a datagram. First, the device checks its memory to determine if it previously learned the MAC address associated with a particular destination IP address. This memory location is referred to as an ARP cache. Because the first occurrence of an IP address means its associated MAC address will not be in the ARP cache, it must learn the MAC address. To do so, the device will broadcast an ARP packet to all devices on the LAN. Exhibit 4.24 illustrates the format of an ARP packet. Note that the numbers shown in some fields in the

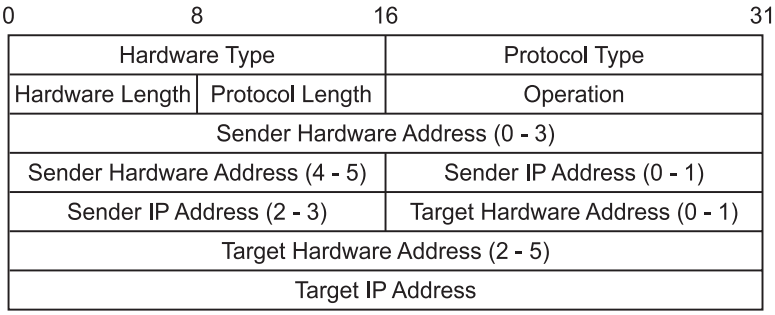


Exhibit 4.24 The ARP Packet Format

ARP packet indicate the byte numbers in a field when a field spans a four-byte boundary.

ARP Packet Fields

To illustrate the operation of ARP, one can examine the fields in the ARP packet. The 16-bit Hardware Type Field indicates the type of network adapter, such as 10 Mbps Ethernet (value = 1), IEEE 802 network (value = 6), etc. The 16-bit Protocol Type Field indicates the protocol for which an address resolution process is being performed. For IP, the Protocol Type Field has a value of hex 0800.

The Hardware Length Field defines the number of bytes in the hardware address. Thus, the ARP packet format can be varied to accommodate different types of address resolutions beyond IP and MAC addresses. Because Ethernet and Token Ring have the same MAC length, the value of this field is 6 for both.

The Protocol Length Field indicates the length of the address for the protocol to be resolved. For IPv4, the value of this field is set to 4. The Operation Field indicates the operation to be performed. This field has a value of 1 for an ARP Request. When a target station responds, the value of this field is changed to 2 to denote an ARP Reply.

The Sender Hardware Address Field indicates the hardware addresses of the station generating the ARP Request or ARP Reply. This field is six bytes in length and is followed by a four-byte Sender IP Address field. The latter indicates the IP address of the originator of the datagram.

The next to last field is the Target Hardware Address Field. Because the ARP process must discover its value, this field is originally set to all zeros in an ARP request. Once a station receives the request and notes it has the same IP address as that in the Target IP Address Field, it places its MAC address in the Target Hardware Address Field. Thus, the last field, Target IP Address, is set to the IP address the originator needs for a hardware address.

Locating the Required Address

To put the pieces together, assume a router receives a datagram from the Internet with the destination address of 205.131.175.5. Further assume that the router has a connection to an Ethernet network, and one station on that network has that IP address. The router needs to determine the MAC address associated with the IP address so it can construct a frame to deliver the datagram. Assuming there is no entry in its ARP cache, the router creates an ARP frame and transmits the frame using a MAC broadcast address of FFFFFFFFFFFF. Because the frame was broadcast to all stations on the network, each device reads the frame. The station that has its protocol stack configured to the same IP address as that of the Target IP Address Field in the ARP frame would respond to the ARP Request. When it does, it will transmit an ARP Reply in which its physical MAC address is inserted into the ARP Target Hardware Address Field that was previously set to zero.

The ARP standard includes provisions for devices on a network to update their ARP table with the MAC and IP address pair of the sender of the ARP Request. Thus, as ARP Requests flow on a LAN, they contribute to the building of tables that reduce the necessity of additional broadcasts.

Gratuitous ARP

There is a special type of ARP referred to as a "gratuitous ARP" that deserves mention. When a TCP/IP stack is initialized, it issues a gratuitous ARP, which represents an ARP request for its own IP address. If the station receives a reply containing a MAC address that differs from its address, this indicates that another device on the network is using its assigned IP address. If this situation occurs, an error message warning of an address conflict will be displayed.

Proxy ARP

A proxy is a device that works on behalf of another device. Thus, a proxy ARP represents a mechanism that enables a device to answer an ARP request on behalf of another device.

The rationale for the development of proxy ARP, which is also referred to as ARP Hack, dates to the early use of subnetting when a LAN could be subdivided into two or more segments. If a station on one segment required the MAC address of a station on another subnet, the router would block the ARP request because it is a Layer 2 broadcast, and routers operate at Layer 3. Because the router is aware of both subnets, it could answer an ARP request on one subnet on behalf of other devices on the second subnet by supplying its own MAC address. The originating device will then enter the router's MAC address in its ARP cache and will correctly transmit packets destined for the end host to the router.

RARP

The Reverse Address Resolution Protocol (RARP) was at one time quite popular when diskless workstations were commonly used in corporations. In such situations, the workstation would know its MAC address, but be forced to learn its IP address from a server on the network. Thus, the RARP would be used by the client to access a server on the local network and would provide the client's IP address. Similar to ARP, RARP is a Layer 2 protocol that cannot normally cross router boundaries. Some router manufacturers implemented RARP, which allows requests and responses to flow between networks.

The RARP frame format is the same as for ARP. The key difference between the two is the setting of field values. The RARP protocol fills in the sender's hardware address and sets the IP address field to zeroes. Upon receipt of the

RARP frame, the RARP server fills in the IP address field and transmits the frame back to the client, reversing the ARP process.

ICMP

This chapter concludes by focusing on the Internet Control Message Protocol (ICMP). If one thinks about IP for a while, one realizes that there is no provision to inform a source of the fact that a datagram encountered some type of problem. This is because one of the functions of ICMP is to provide a messaging capability that reports different types of errors that can occur during the processing of datagrams. In addition to providing an error reporting mechanism, ICMP includes certain types of messages that provide a testing capability.

Overview

ICMP messages are transmitted within an IP datagram as illustrated in Exhibit 4.25. Note that although each ICMP message has its own format, they all begin with the same three fields. Those fields are an eight-bit Type Field, an eight-bit Code Field, and a 16-bit Checksum Field.

One can obtain familiarity with the capability of ICMP by examining the use of some of the fields within an ICMP message. The Type and Code Fields within an ICMP message are discussed first.

The ICMP Type Field

The purpose of the ICMP Type Field is to define the meaning of the message as well as its format. Two of the most popularly used ICMP messages use

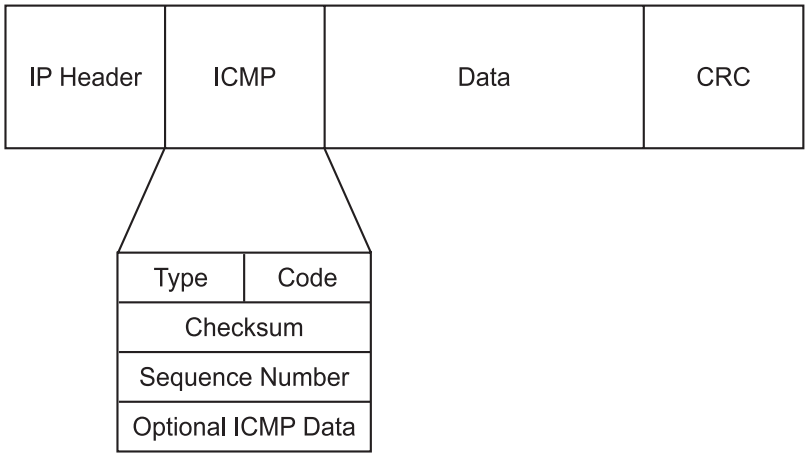


Exhibit 4.25 ICMP Messages are Transported via Encapsulation within an IP Datagram

type values of 0 and 8. A Type Field value of 8 represents an echo request, while a Type Field value of 0 denotes an ICMP echo reply. Although their official names are Echo Request and Echo Reply, most people are more familiar with the term Ping, which is used to reference both the request and the reply. Exhibit 4.26 lists ICMP Type Field values that currently identify specific types of ICMP messages.

Exhibit 4.26 ICMP Type Field Values

Type	*Name*
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo Request
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20–29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41–255	Reserved

The ICMP Code Field

The ICMP Code Field provides additional information about a message defined in the Type Field. The Code Field may not be meaningful for certain ICMP messages. For example, both Type Field values of 0 (echo reply) and 8 (echo request) always have a Code Field value of 0. In comparison, a Type Field value of 3 (Destination Unreachable) can have one of 16 possible Code Field values, which further defines the problem. Exhibit 4.27 lists the Code Field values presently assigned to ICMP messages based upon their Type Field values.

Evolution

Over the years from its first appearance in RFC 792, ICMP has evolved through the addition of many functions. For example, a Type 4 (source quench) represents the manner by which an endstation indicates to the originator of a message that the host cannot accept the rate at which the originator is transmitting packets. The recipient will send a flow of ICMP Type 4 messages to the originator as a message for the origination to slow down its transmission. When an acceptable flow level is reached, the recipient will terminate its generation of source quench messages. Although popularly used many years ago for controlling traffic, the TCP slow-start algorithm has superseded a majority of the use of ICMP Type 4 messages.

ICMP message types that warrant discussion are Type 5 and Type 7. A router generates a Type 5 (redirect) message when it receives a datagram and determines that there is a better route to the destination network. This ICMP message informs the sender of the better route. A Type 7 message (time exceeded) indicates that the Time to Live Field value in an IP datagram header was decremented to 0, and the datagram was discarded. As will be discussed later in this book, ICMP provides a foundation for several diagnostic testing applications. Unfortunately, this testing capability can be abused by unscrupulous persons and results in many organizations filtering ICMP messages so that they do not flow from the Internet onto a private network.

Exhibit 4.27 ICMP Code Field Values Based on Message Type

Message Type	*Code Field Values*
3	Destination Unreachable **Codes** 0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated 9 Communication with Destination Network is Administratively Prohibited 10 Communication with Destination Host is Administratively Prohibited 11 Destination Network Unreachable for Type of Service 12 Destination Host Unreachable for Type of Service 13 Destination Host Unreachable for Type of Service 14 Communication Administratively Prohibited 15 Precedence cutoff in effect
5	Redirect **Codes** 0 Redirect Datagram for the Network (or subnet) 1 Redirect Datagram for the Host 2 Redirect Datagram for the Type of Service and Network 3 Redirect Datagram for the Type of Service and Host
6	Alternate Host Address **Codes** 0 Alternate Address for Host
11	Time Exceeded **Codes** 0 Time to Live Exceeded in Transit 1 Fragment Reassembly Time Exceeded
12	Parameter Problem **Codes** 0 Point Indicates the Error 1 Missing a Required Option 2 Bad Length

(continues)

Exhibit 4.27 ICMP Code Field Values Based on Message Type *(continued)*

Message Type	*Code Field Values*
40	Photuris **Codes** 0 Reserved 1 Unknown security parameters index 2 Valid security parameters, but authentication failed 3 Valid security parameters, but decryption failed

Chapter 5

The Transport Layer

The purpose of this chapter is to acquaint the reader with the two transport layer protocols supported by the ICP/IP suite. These protocols are the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

As indicated in Chapter 3, either TCP or UDP can be identified by the setting of an applicable value in the IP header. Although the use of either protocol results in the placement of the appropriate transport layer header behind the IP header, there are significant differences between the functionality of each transport protocol. Those differences make one protocol more suitable for certain applications than the other protocol, and vice versa.

TCP

The Transmission Control Protocol (TCP) is a connection-oriented protocol. This means that the protocol will not forward data until a session is established in which the destination acknowledges it is ready to receive data. This also means that the TCP setup process requires more time than when UDP is used as the transport layer protocol. However, because one would not want to commence certain operations, such as remote log on or a file transfer, unless one knows the destination is ready to support the appropriate application, the use of TCP is more suitable than UDP for certain applications. Conversely, in examining UDP, one will realize that this transport layer protocol similarly supports certain applications better than other applications. The best way to become familiar with TCP is by first examining the fields in its header.

The TCP Header

The TCP header consists of 12 fields as illustrated in Exhibit 5.1. In comparing TCP and UDP, one realizes that the TCP header is far more complex. The

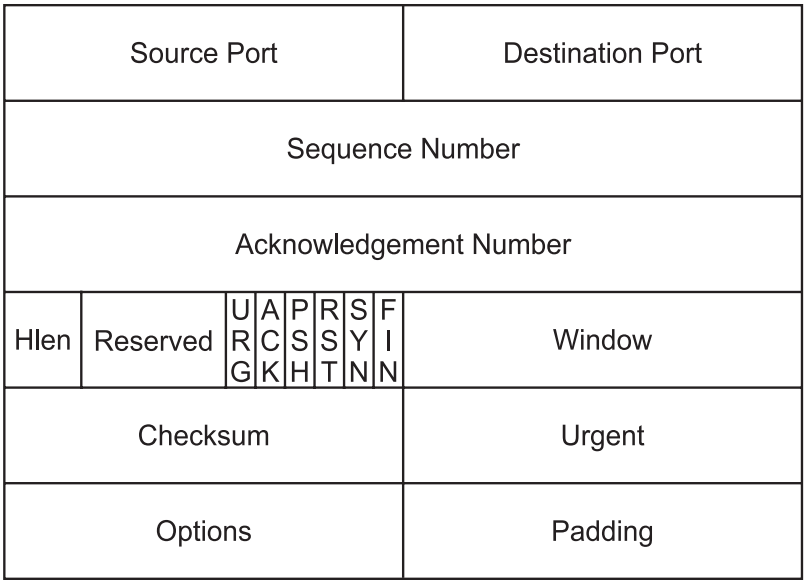


Exhibit 5.1 The TCP Header

reason for this additional complexity results from the fact that TCP is not only a connection-oriented protocol, but, in addition, supports error detection and correction as well as packet sequencing, with the latter used to note the ordering of packets and includes determining if one or more packets are lost.

Source and Destination Port Fields

The source and destination port fields are each 16 bits in length. Each field denotes a particular process or application. In actuality, most applications use the destination port number to denote a particular process or application, and either set the source port field value to a random number greater than 1024 or to zero. The reason the destination port number defines the process or application results from the fact that an application operating at the receiver normally operates acquiescently, waiting for requests, looking for a specific destination port number to determine the request.

The reason the originator sets the source port to zero or a value above 1023 is due to the fact that the first 1023 out of 65,536 available port numbers are standardized with respect to the type of traffic transported via the use of specific numeric values. To illustrate the use of port numbers, assume one station wishes to open a Telnet connection with a distant server. Because Telnet is defined as port 23, the application will set the destination port value to that numeric. The source port will normally be set to a random value above 1023 and an IP header will then add the destination and source IP addresses for routing the datagram from the client to the server. In some literature, one

may encounter the term "socket," sometimes incorrectly used as a synonym for port. In actuality, the destination port in the TCP or UDP header plus the destination IP address cumulatively identify a unique process or application on a host. The combination of port number and IP address is correctly referenced as a socket. At the server, the destination port value of 23 identifies the application as Telnet. When the server forms a response, it first reverses source and destination IP addresses. Similarly, the server places the source port number in the destination port field, which enables the Telnet originator's application to correctly identify the response to its initial datagram.

Multiplexing and Demultiplexing

Port numbers play an important role in TCP/IP as they enable multiple applications to flow between the same pair of stations or from multiple nonrelated stations to a common station. This flow of multiple applications to a common address is referred to as multiplexing. Upon receipt of a datagram, the removal of the IP and TCP headers requires the remaining portion of the packet to be routed to its correct process or application, based on the destination port number in the TCP header. This process is referred to as demultiplexing.

Both TCP and UDP use port numbers to support the multiplexing of different processes or applications to a common IP address. An example of this multiplexing and demultiplexing of packets is illustrated in Exhibit 5.2. The top left portion of Exhibit 5.2 illustrates how both Telnet and FTP, representing two TCP applications, can be multiplexed into a stream of IP datagrams that flow to a common IP address. In comparison, the top right portion of Exhibit 5.2 illustrates how, through port numbering, UDP ports permit a similar method of multiplexing of applications.

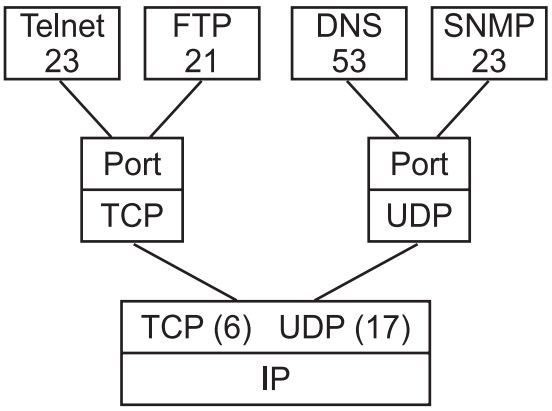


Exhibit 5.2 Port Numbers with Multiple Applications Multiplexed via Serial Communications to a Common IP Address

Port Numbers

The "universe" of both TCP and UDP port numbers can vary from a value of 0 to 65535, resulting in a total of 65,535 ports capable of being used by each transport protocol. This so-called port universe is divided into three ranges referred to as Well-Known Ports, Registered Ports, and Dynamic or Private Ports.

Well-Known Ports

Well-Known Ports are the most commonly used port values as they represent assigned numerics that identify specific processes or applications. Ports 0 through 1023 represent the range of Well-Known Ports. These port numbers are assigned by the Internet Assigned Numbers Authority (IANA) and are used to indicate the transportation of standardized processes. Where possible, the same Well-Known Port number assignments are used with TCP and UDP. Ports used with TCP are normally used to provide connections that transport long-term conversations. In some literature, one may encounter Well-Known Port numbers being specified as in the range of value from 0 to 255. While this range was correct many years ago, the modern range for assigned ports managed by the IANA was expanded to cover the first 1024 port values from 0 to 1023. Exhibit 5.3 provides a summary of the port value assignments from 0 through 255 for Well-Known Ports to include the service supported by a particular port and the type of port — TCP or UDP — for which the port number is primarily used. A good source for the full list of assigned port numbers is RFC 1700.

Registered Ports

Registered ports are ones whose values range from 1024 through 49151. Although all ports above 1023 can be used freely, the IANA requests vendors to register their application port numbers with them.

Dynamic or Private Ports

The third range of port numbers is from 49152 through 65535. This port number range is associated with dynamic or private ports. This port range is usually used by new applications that remain to be standardized, such as Internet telephony.

Sequence and Acknowledgment Number Fields

TCP is a byte-oriented sequencing protocol. Thus, a sequence field is necessary to ensure that missing or misordered packets are noted or identified. That field is 32 bits in length and provides the mechanism for ensuring that missing or misordered packets are noted or identified.

Exhibit 5.3 Well-Known TCP and UDP Services and Port Use

Keyword	*Service*	*Port Type*	*Port Number*
TCPMUX	TCP Port Service Multiplexer	TCP	1
RJE	Remote Job Entry	TCP	5
ECHO	Echo	TCP and UDP	7
DAYTIME	Daytime	TCP and UDP	13
QOTD	Quote of the Day	TCP	17
CHARGEN	Character Generator	TCP	19
FTD-DATA	File Transfer (Default Data)	TCP	20
FTP	File Transfer (Control)	TCP	21
TELNET	Telnet	TCP	23
SMTP	Simple Mail Transfer Protocol	TCP	25
MSG-AUTH	Message Authentication	TCP	31
TIME	Time	TCP	37
NAMESERVER	Host Name Server	TCP and UDP	42
NICNAME	Who Is	TCP	43
DOMAIN	Domain Name Server	TCP and UDP	53
BOOTPS	Bootstrap Protocol Server	TCP	67
BOOTPC	Bootstrap Protocol Client	TCP	68
TFTP	Trivial File Transfer Protocol	UDP	69
FINGER	Finger	TCP	79
HTTP	World Wide Web	TCP	80
KERBEROS	Kerberos	TCP	88
RTELNET	Remote Telenet Service	TCP	107
POP2	Post Office Protocol Version 2	TCP	109
POP3	Post Office Protocol Version 3	TCP	110
NNTP	Network News Transfer Protocol	TCP	119
NTP	Network Time Protocol	TCP and UDP	123
NETBIOS-NS	NetBIOS Name Server	UDP	137
NETBIOS-DGM	NetBIOS Datagram Service	UDP	138
NETBIOS-SSN	NetBIOS Session Service	UDP	139
NEWS	News	TCP	144
SNMP	Simple Network Management Protocol	UDP	161
SNMPTRAP	Simple Network Management Protocol Traps	UDP	162
BGP	Border Gateway Protocol	TCP	179
HTTPS	Secure HTTP	TDP	413
RLOGIN	Remote Login	TCP	513
TALK	Talk	TCP and UDP	517

The actual entry in the Sequence Number field is based on the number of bytes in the TCP Data field. That is, because TCP was developed as a byte-oriented protocol, each byte in each packet is assigned a sequence number. Because it would be most inefficient for TCP to transmit one byte at a time, groups of bytes, typically 512 or 536, are placed in a segment and one sequence number is assigned to the segment and placed in the Sequence field. That number is based on the number of bytes in the current segment as well as previous segments, as the Sequence field value increments its count until all 16-bit positions are used and then continues via a rollover through zero. For example, assume the first TCP segment contains 512 bytes and a second segment will have the sequence number 1024.

The Acknowledgment Number field, which is also 32 bits in length, is used to verify the receipt of data. The number in this field also reflects bytes. For example, returning to the example sequence of two 512-byte segments, when the first segment is received, the receiver expects the next sequence number to be 513. Therefore, if the receiver were acknowledging each segment, it would first return an acknowledgment with a value of 513 in the Acknowledgment Number field. When it acknowledges the next segment, the receiver would set the value in the Acknowledgment Number field to 1025, etc.

Because it would be inefficient to have to acknowledge each datagram, a variable or "sliding" window is supported by TCP. That is, returning an Acknowledgment Number field value of n + 1 would indicate the receipt of all bytes through byte n. If the receiver has the ability to process a series of multiple segments and each is received without error, it would be less efficient to acknowledge each datagram. Thus, a TCP receiver can process a variable number of segments prior to returning an acknowledgment that informs the transmitter that n bytes were received correctly. To ensure lost datagrams or lost acknowledgments do not place TCP in an infinite waiting period, the originator sets a timer and will retransmit data if it does not receive a response within a predefined period of time.

The previously described use of the Acknowledgment Number field is referred to as Positive Acknowledgment Retransmission (PAR). Under PAR, each unit of data must be either implicit (sending a value of n + 1 to acknowledge receipt of n bytes) or explicit. If a unit of data is not acknowledged by the time the originator's timeout period is reached, the previous transmission is retransmitted. When the Acknowledgment Number field is in use, a flag bit, referred to as the ACK flag in the Code field, will be set. The six bit positions in the Code Bits field are discussed below.

Hlen Field

The Header Length (Hlen) field is four bits in length. This field, which is also referred to as the Offset field, contains a value that indicates where the TCP header ends and the Data field begins. This value is specified as a number of 32-bit words. It is required due to the fact that the inclusion of options can result in a variable-length header. Because the minimum length of the

TCP header is 20 bytes, the minimum value of the Hlen field would be 5, denoting five 32-bit words, which equals 20 bytes.

Code Bits Field

As previously indicated in Exhibit 5.1, there are six individual one-bit fields within the Code Bits field. Each bit position functions as a flag to indicate whether or not a function is enabled or disabled. Thus, to obtain an appreciation for the use of the Code Bits field, one should examine each bit position in that field.

URG Bit

The Urgent (URG) bit or flag is used to denote an urgent or priority activity. When such a situation occurs, an application will set the URG bit position, which acts as a flag and results in TCP immediately transmitting everything it has for the connection instead of waiting for additional characters. An example of an action that could result in an application setting the Urgent flag would be a user pressing the CTRL-BREAK key combination.

A second meaning resulting from the setting of the Urgent bit or flag is that it also indicates that the Urgent Pointer field is in use. Here, the Urgent Pointer field indicates the offset in bytes from the current sequence number where the Urgent data is located.

ACK Bit

The setting of the ACK bit indicates that the segment contains an acknowledgment to a previously transmitted datagram or series of datagrams. Then the value in the Acknowledgment Number field indicates the correct receipt of all bytes through byte n by having the byte number n + 1 in the field.

PSH Bit

The third bit position in the Code Bit field is the Push (PSH) bit. This one-bit field is set to request the receiver to immediately deliver data to the application and flags any buffering. Normally, the delivery of urgent information would result in the setting of both the URG and PSH bits in the Code Bits field.

RST Bit

The fourth bit position in the Code Bits field is the Reset (RST) bit. This bit position is set to reset a connection. By responding to a connection request with the RST bit set, this bit position can also be used as a mechanism to decline a connection request.

SYN Bit

The fifth bit in the Code Bits field is the Synchronization (SYN) bit. This bit position is set at start-up during what is referred to as a three-way handshake.

FIN Bit

The sixth and last bit position in the Code Bits field is the Finish (FIN) bit. This bit position is set by the sender to indicate that it has no additional data, and the connection should be released.

Window Field

The Window field is 16 bits in length and provides TCP with the ability to regulate the flow of data between source and destination. Thus, this field indirectly performs flow control.

The Window field indicates the maximum number of bytes that the receiving device can accept. Thus, it indirectly indicates the available buffer memory of the receiver. Here, a large value can significantly improve TCP performance as it permits the originator to transmit a number of segments without having to wait for an acknowledgment while permitting the receiver to acknowledge the receipt of multiple segments with one acknowledgment.

Because TCP is a full-duplex transmission protocol, both the originator and recipient can insert values in the Window field to control the flow of data in each direction. By reducing the value in the Window field, one end of a session in effect informs the other end to transmit less data. Thus, the use of the Window field provides a bi-directional flow control capability.

Checksum Field

The Checksum field is 16 bits (or two bytes) in length. The function of this field is to provide an error detection capability for TCP. To do so, this field is primarily concerned with ensuring that key fields are validated instead of protecting the entire header. Thus, the checksum calculation occurs over what is referred to as a 12-byte pseudo-header. This pseudo-header includes the 32-bit Source and Destination Address fields in the IP header, the eight-bit Protocol field, and a Length field that indicates the length of the TCP header and data transported within the TCP segment. Thus, the primary purpose of the Checksum field is to ensure that data has arrived at its correct destination, and the receiver has no doubt about the address of the originator nor the length of the header and the type of application data transported.

Urgent Pointer Field

The Urgent Pointer field is one byte in length. The value in this field acts as a pointer to the sequence number of the byte following the urgent data. As

previously noted, the URG bit position in the Code field must be set for the data in the Urgent Pointer field to be interpreted.

Options

The Options field, if present, can be variable in length. The purpose of this field is to enable TCP to support various options, with Maximum Segment Size (MSS) representing a popular TCP option. Because the header must end on a 32-bit boundary, any option that does not do so is extended via pad characters that in some literature is referred to as a Padding field.

Padding Field

The Padding field is optional and is included only when the Options field does not end on a 32-bit boundary. Thus, the purpose of the Padding field is to ensure that the TCP header, when extended, falls on a 32-bit boundary.

Given an appreciation for the composition of the TCP header, one can now focus on the manner by which this protocol operates. In doing so, the reader will examine how TCP establishes a connection with a distant device and its initial handshaking process, its use of sequence and acknowledgment numbers, how flow control is supported by the protocol, and how the protocol terminates a session.

Connection Establishment

As mentioned earlier in this section, TCP is a connection-oriented protocol that requires a connection between two stations to be established prior to the actual transfer of data occurring. The actual manner by which an application communicates with TCP is through a series of function calls. To obtain an appreciation for the manner by which TCP establishes a session, one must first examine connection function calls used by applications, for example, Telnet and FTP.

Connection Function Calls

Exhibit 5.4 illustrates the use of the OPEN connection function calls during the TCP connection establishment process. This process commences when an application requires a connection to a remote station. At that time, the application will request TCP to place an OPEN function call. There are two types of OPEN function calls: passive and active. A passive OPEN function call represents a call to allow connections to be accepted from a remote station. This type of call is normally issued upon application start-up, informing TCP that, for example, FTP or Telnet is active and ready to accept connections originating from other stations. TCP will then note that the application is active, note its port assignment, and then allow connections on that port number.

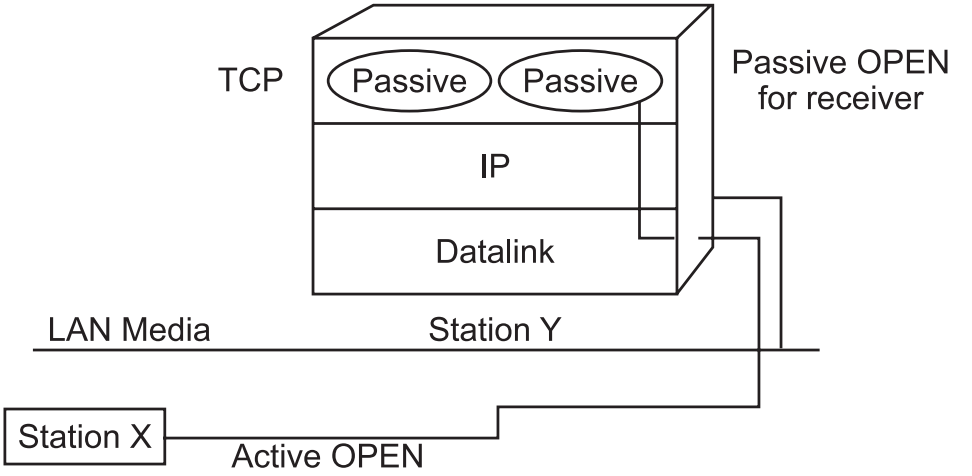


Exhibit 5.4 Using Function Calls to Establish a TCP Connection

Port Hiding

One of the little-known aspects of TCP is the fact that some organizations attempt to hide their applications by configuring applications for ports other than well-known ports. For example, assigning Telnet to port 2023 instead of port 23 is an example of port hiding. Although a person with port scanning software would be able to easily discover that port 2023 is being used, the theory behind port hiding is that it reduces the ability of lay personnel to easily discover applications at different network addresses and then attempt to use those applications.

Passive OPEN

Returning to the use of a passive OPEN function call, its use governs the number of connections allowed. That is, while a client would usually issue one passive OPEN, a server would issue multiple OPENs because it is designed to service multiple sessions. Another term used for the passive end of the TCP action is responder or TCP responder. Thus, a TCP responder can be thought of as an opening up of connection slots to accept any inbound connection request without waiting for any particular station request.

Active OPEN

A station that needs to initiate a connection to a remote station issues the second type of OPEN call. This type of function call is referred to as an active OPEN. In the example illustrated in Exhibit 5.4, station X would issue an active OPEN call to station Y. For the connection to be serviced by station Y, that station must have previously issued a passive OPEN request which, as previously

explained, allows incoming connections to be established. To successfully connect, station X's active OPEN must use the same port number that the passive OPEN used on station Y. In addition to active and passive OPEN calls, other calls include CLOSE (to close a connection), SEND and RECEIVE (to transfer information), and STATUS (to receive information for a previously established connection).

Given an appreciation for the use of active and passive OPEN calls to establish a TCP connection, one can now explore the manner by which TCP segments are exchanged. The exchange of segments enables a session to occur. The initial exchange of datagrams that transport TCP segments is referred to as a three-way handshake. It is important to note how and why this process occurs. It has been used in modified form as a mechanism to create a denial-of-service (DoS) attack, which is discussed in Chapter 9.

The Three-Way Handshake

To ensure that the sender and receiver are ready to commence the exchange of data requires that both parties to the exchange be synchronized. Thus, during the TCP initialization process, the sender and receiver exchange a few control packets for synchronization purposes. This exchange is referred to as a three-way handshake. This functions as a mechanism to synchronize each endpoint at the beginning of a TCP connection with a sequence number and an acknowledgment number.

Overview

A three-way handshake begins with the originator sending a segment with its SYN bit in the Code Bit field set. The receiving station will respond with a similar segment with its ACK bit in the Code Bit field set. Thus, an alternate name for the three-way handshake is an "initial SYN-SYN-ACK" sequence.

Operation

To illustrate the three-way handshake, one can continue from the prior example shown in Exhibit 5.4, in which station X placed an active OPEN call to TCP to request a connection to a remote station and an application on that station. Once the TCP/IP protocol stack receives an active OPEN call, it will construct a TCP header with the SYN bit in the Code Bit field set. The stack will also assign an initial sequence number and place that number in the Sequence Number field in the TCP header. Other fields in the header, such as the destination port number, are also set and the segment is then transferred to IP for the formation of a datagram for transmission onto the network.

To illustrate the operation of the three-way handshake, consider Exhibit 5.5 which illustrates the process between stations X and Y. Because the initial sequence number does not have to start at zero, assume it commenced at

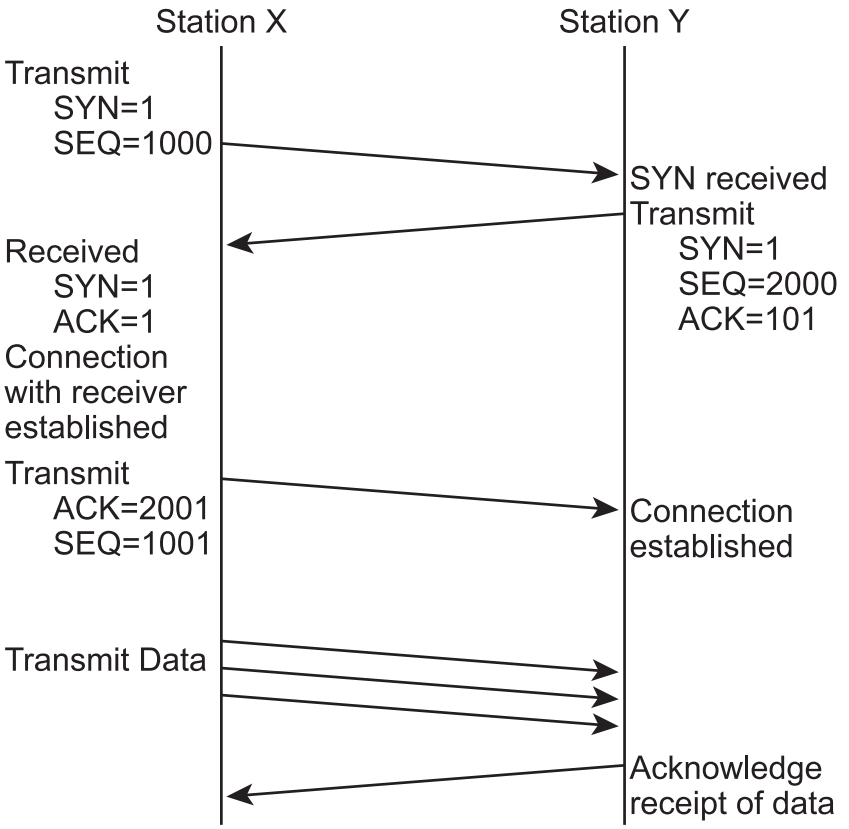


Exhibit 5.5 The Three-Way Handshake

1000 and then further assume that the value was placed in the Sequence Number field. Thus, the TCP header flowing from station X to station Y is shown with SYN = 1 and SEQ = 1000.

Because the IP header results in the routing of a datagram to station Y, that station strips the IP header and notes that the setting of the SYN bit in the TCP header represents a connection request. Assuming station Y can accept a new connection, it will acknowledge the connection request by building a TCP segment. That segment will have its SYN and ACK bits in its Code Bit field set. In addition, station Y will place its own initial sequence number in the Sequence Number field of the TCP header it is forming. Because the connection request had a sequence number of 1000, station Y will acknowledge receipt by setting its Acknowledgment field value to 1001 (station X sequence number plus 1), which indicates the next expected sequence number.

Once station Y forms its TCP segment, the segment has an IP header added to form a datagram. The datagram flows to station X. Station X receives the datagram, removes the IP header, and notes via the setting of the XYN and ACK bits and Sequence Number field value that it is a response to its previously issued connection request. To complete the connection request, station X must, in effect, acknowledge the acknowledgment. To do so, station X will

construct a new TCP segment in which the ACK bit will be set and the sequence number will be incremented by 1 to 1001. Station X will also set the acknowledgment number to 2001 and form a datagram that is transmitted to station Y. Once station Y examines the TCP header and confirms the correct values for the Acknowledgment and Sequence Number fields, the connection becomes active. At this point in time, both data and commands can flow between the two endpoints. As this occurs, each side of the connection maintains its own set of tables for transmitted and received sequence numbers. Those numbers are always in ascending order. When the applicable 16-bit field reaches its maximum value, the settings wrap to 0.

In examining the three-way handshake illustrated in Exhibit 5.5, note that after the originating station establishes a connection with the receiver, it transmits a second TCP initialization segment to the receivers and follows that segment with one or more IP datagrams that transport the actual data. In Exhibit 5.5, a sequence of three datagrams is shown being transmitted prior to station Y, generating an acknowledgment to the three segments transported in the three datagrams. The actual number of outstanding segments depends on the TCP window, discussed next.

The TCP Window

TCP is a connection-oriented protocol that includes a built-in capability to regulate the flow of information, a function referred to as flow control. TCP manages the flow of information by increasing or decreasing the number of segments that can be outstanding at any point in time. For example, under periods of congestion when a station is running out of available buffer space, the receiver may indicate it can only accept one segment at a time and delay its acknowledgment to ensure it can service the next segment without losing data. Conversely, if a receiver has free and available buffer space, it may allow multiple segments to be transmitted to it and quickly acknowledge the segments.

TCP forms segments sequentially in memory. Each segment of memory waits for an IP header to be added to form a datagram for transmission. A "window" is placed over this series of datagrams that structures three types of data: data transmitted and acknowledged; data transmitted, but not yet acknowledged; and data waiting to be transmitted. Because this "window" slides over the three types of data, the window is referred to as a "sliding window."

Exhibit 5.6 illustrates the use of the TCP sliding window for flow control purposes. Although the actual TCP segments size is normally 512 bytes, for simplicity of illustration, a condensed sequence of segments with sequence numbers varying by unity are shown. In this example, assume that sequence numbers 10 through 15 have been transmitted to the destination station. The remote station acknowledges receipt of those segments. Datagrams containing segment sequence numbers 16 through 20 were transmitted by the source station, but at this particular point in time have not received an acknowledgment. Thus, that data represents the second type of data covered by a sliding window.

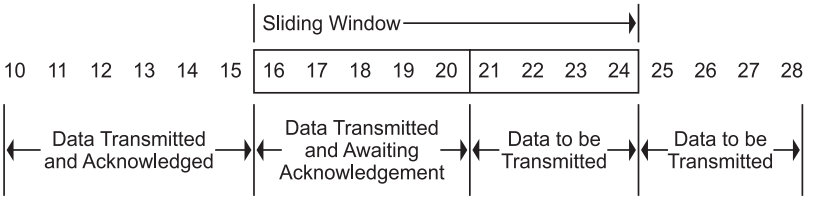

Sliding Window
10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
Data Transmitted and Acknowledged
Data Transmitted and Awaiting Acknowledgement
Data to be Transmitted
Data to be Transmitted

Exhibit 5.6 Flow Control and the TCP Sliding Window

Note that this window will slide up the segments as each datagram is transmitted. The third type of data covered by the sliding window is segments. In Exhibit 5.6, segments 21 through 24 are in the source station awaiting transmission, while segments 25 through 28 are awaiting coverage by the sliding window.

Returning to Exhibit 5.1, which illustrated the TCP header, note the field labeled "Window." That field value indirectly governs the length of the sliding window. In addition, the setting of that field provides a flow control mechanism. For example, the Window field transmitted by a receiver to a sender indicates the range of sequence numbers, which equates to bytes, that the receiver is willing to accept. If a remote station cannot accept additional data, it would then set the Window field value to 0. The receiving station continues to transmit TCP segments with the Window field set to 0 until its buffer is emptied a bit, no pun intended, in effect allowing the resumption of transmission conveying data by the originator. That is, when the transmitting station receives a response with a Window field value of zero, it replies to the response with an ACK (Code field ACK bit set to 1) and its Window field set to a value of 0. This inhibits the flow of data. When sufficient buffer space becomes available at the receiver, it will form a segment with its Window field set to a non-zero value, an indication that it can again receive data. At this point, the transmitting of data goes to the receiver.

Avoiding Congestion

One of the initial problems associated with TCP is the fact that a connection could commence with the originator transmitting multiple segments, up to the Window field value returned by the receiver during the previously described three-way handshake. If there are slow-speed WAN connections between originator and recipient, it is possible that routers could become saturated when a series of transmissions originated at the same time. In such a situation, the router would discard datagrams, causing retransmissions that continued the abnormal situation. The solution developed to avoid this situation is referred to as a TCP slow start process.

TCP Slow Start

Slow start represents an algorithm procedure added to TCP that implicitly uses a new window, referred to as the congestion window. This window is not

contained as a field in the TCP header. Instead, it becomes active through the algorithm that defined the slow start process. That is, when a new connection is established, the congestion window is initialized to a size of one segment, typically 512 or 536 bytes. Each time an ACK is received, the congestion window's length is increased by one segment. The originator can transmit any number of segments up to the minimum value of the congestion window or the Window field value (Advertised Window). Note that flow control is imposed by the transmitter in one direction through the congestion window, while it is imposed in the other direction by the receiver's advertised Window field value.

Although slow start commences with a congestion window of one segment, it builds up exponentially until it reaches the Advertised Window size. That is, it is incremented by subsequent ACKs from 1 to 2, then it is increased to 4, 8, 16, etc. until it reaches the Advertised Window size. Once this occurs, segments are transferred using the Advertised Window size for congestion control and the slow start process is terminated.

The Slow Start Threshold

In addition to working at initiation, slow start will return upon the occurrence of one of two conditions: duplicate ACKs, or a timeout condition where a response is not received within a predefined period of time. When either situation occurs, the originator commences another algorithm referred to as the congestion control algorithm.

When congestion occurs, a comparison is initiated between the congestion window size and the current advertised window size. The smaller number is halved and saved in a variable referred to as a slow start threshold. The minimum value of the slow start threshold is 2 segments unless congestion occurred via a timeout, with the congestion window then set to a value of 1, the same as a slow start process. The TCP originator has the option of using the slow start start-up or congestion avoidance. To determine which method to use, the originator compares the congestion value to the value of the slow start threshold. If the congestion value matches the value of the slow start threshold, the congestion avoidance algorithm will be used. Otherwise, the originator will use the slow start method. Having previously described the slow start method, the focus shifts to the congestion avoidance method and to the algorithm it uses.

Upon the receipt of ACKs, the congestion window will be increased until its value matches the value saved in the slow start threshold. When this occurs, the slow start algorithm terminates and the congestion avoidance algorithm starts. This algorithm multiplies the segment size by two, divides that value by the congestion window size, and then continually increases its value based on the previously described algorithm each time an ACK is received. The result of this algorithm is a more linear growth in the number of segments that can be transmitted in comparison to the exponential growth of the slow start algorithm.

TCP Retransmissions

While it is obvious that the negative acknowledgment of a segment by the receiver returning the same segment number expected indicates a retransmission request, what happens if a datagram is delayed? Because delays across a TCP/IP network depend on the activity of other routers in the network, the number of hops in the path between source and destination, and other factors, it is relatively impossible to have an exact expected delay prior to a station assuming data is lost and retransmitting. Recognizing this situation, the developers of TCP included an adaptive retransmission algorithm in the protocol. Under this algorithm, when TCP submits a segment for transmission, the protocol records the segment sequence number and time. When an acknowledgment is received to that segment, TCP also records the time, obtaining a round-trip delay. TCP uses such timing information to construct an average round-trip delay that is used by a timer to denote, when the timer expires, that a retransmission should occur. When a new transmit-response sequence occurs, another round-trip delay is computed, which slightly changes the average. Thus, this technique slowly changes the timer value that governs the acceptable delay for waiting for an ACK. With an understanding of how TCP determines when to retransmit a segment, coverage of this protocol concludes with how TCP gracefully terminates a session.

Session Termination

If one remembers the components of the Code Bit field, one remembers that this field has a FIN bit. The purpose of this bit is to enable TCP to gracefully terminate a session. Before TCP supports full-duplex communication, each party to the session must close the session. This means that both the originator and recipient must exchange segments with the FIN bit set in each segment.

Exhibit 5.7 illustrates the exchange of segments to gracefully terminate a TCP connection. In this example, assume station X has completed its transmission and indicates this fact by sending a segment to station Y with the FIN bit set. Station Y will acknowledge the segment with an ACK. At this point in time, station Y will no longer accept data from station X. Station Y can continue to accept data from its application to transmit to station X. If station Y does not have any more data to transmit, it will then completely close the connection by transmitting a segment to station X with the FIN bit set in the segment. Station X will then ACK that segment and terminate the connection. If an ACK should be lost in transit, segments with FIN are transmitted and a timer is set. Then either an ACK is received or a timeout occurs, which serves to close the connection.

UDP

The User Datagram Protocol (UDP) is the second transport layer protocol supported by the TCP/IP protocol suite. UDP is a connectionless protocol.

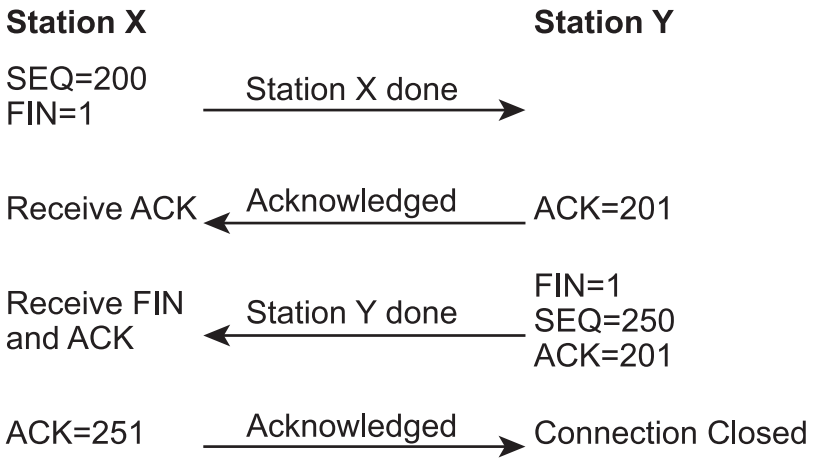


Exhibit 5.7 Terminating a TCP Connection

This means that an application using UDP can have its data transported in the form of IP datagrams without first having to establish a connection to the destination. This also means that when transmission occurs via UDP, there is no need to release a connection, simplifying the communications process. Other features of UDP include the fact that this protocol has no ordering capability, nor does it provide an error detection and correction capability. This in turn results in a header that is greatly simplified and is much smaller than that of TCPs.

The UDP Header

Exhibit 5.8 illustrates the composition of the UDP header. This header consists of 64 bytes, followed by actual user data. In comparing the TCP and UDP headers, it is easy to note the relative simplicity of the latter because it lacks many of the features of the former. For example, because it does not require the acknowledgment of datagrams nor sequence datagrams, there is no need for Sequence and Acknowledgment fields. Similarly, because UDP does not provide a flow control mechanism, the TCP Window field is removed. The result of UDP performing a best-effort delivery mechanism is a relatively small transport layer protocol header, with the protocol relatively simple in

Source Port	Destination Port
Message Length	Checksum

Exhibit 5.8 The User Datagram Protocol Header

comparison to TCP. The best way to understand the operation of UDP is via an examination of its header. Note that similar to TCP, an IP header will prefix the UDP header, with the resulting message consisting of the IP header, UDP header, and user data referred to as a UDP datagram.

Source and Destination Port Fields

The Source and Destination Port fields are each 16 bits (or two bytes) in length and function in a manner similar to their counterparts in the TCP header. That is, the Source Port field is optionally used, with a value either randomly selected or filled in with zeroes when not in use, while the destination port contains a numeric that identifies the destination application or process.

Length Field

The Length field indicates the length of the UDP datagram, to include header and user data that follows the header. This two-byte field has a minimum value of 8, which represents a UDP header without data.

Checksum Field

The Checksum field is two bytes in length. The use of this field is optional and its value is set to 0 if the application does not require a checksum. If a checksum is required, it is calculated on what is referred to as a pseudo-header. The pseudo-header is a logically formed header that consists of the Source and Destination addresses and the Protocol field from the IP header. By verifying the contents of the two address fields through its checksum computation, the pseudo-header ensures that the UDP datagram is delivered to the correct destination network and host on the network. This does not verify the contents of the datagram.

Operation

Because the UDP header does not include within the protocol an acknowledgment capability or a sequence numbering capability, it is up to the application layer to provide this capability. This enables some applications to add this capability, whereas other applications that run on top of UDP may elect not to include one or both. As previously described, a UDP header and its data are prefixed with an IP header to form a data frame. Upon receipt of the datagram, the IP layer strips off that header and submits the remainder to UDP software at the transport layer. The UDP layer reads the destination port number as a mechanism to demultiplex the data and send it to its appropriate application.

Applications

UDP is primarily used by applications that transmit relatively short segments and for which the use of TCP would result in a high level of overhead in comparison to UDP. Common examples of applications that use UDP as a transport protocol include the Simple Network Management Protocol (SNMP), Domain Name Service (DNS), and the newly emerging series of applications from numerous vendors that transport digitized voice over the Internet and are collectively referred to as Internet telephony. Concerning Internet telephony, most implementations applications use both TCP and UDP. TCP is used for call setup, whereas UDP is used to transport digitized voice once the setup operation is completed. Because real-time voice cannot tolerate more than a fraction of a second of delay, Internet applications do not implement error detection and correction, as retransmissions would add delays that would make reconstructed voice sound awkward. Instead, because voice does not rapidly change, applications may either "smooth" an error or drop the datagram and generate a small period of noise that cannot affect the human ear. This is because most Internet telephony applications transmit 10-ms or 20-ms slices of digitized voice, making the error or even the loss of one of a few datagrams transmitting such slices of a conversation most difficult to notice.

Chapter 6

Applications and Built-in Diagnostic Tools

The TCP/IP protocol suite includes a number of built-in diagnostic tools that developers provide as associated applications running under the operation system that supports the suite. Thus, the primary focus of this chapter is on a core set of applications that can be used to obtain insight into the flow of data across a TCP/IP network. Through the use of the application programs discussed in this chapter, one can determine if the protocol stack is operating correctly on a host, whether or not a host is reachable via a network, and the delay or latency between different networks with respect to the flow of data from one network to another. Because knowledge of the Domain Name System (DNS) is important to obtain an understanding of the operation and constraints associated with different applications that provide a diagnostic testing capability, an overview of DNS is given in the first section in this chapter. Once this is accomplished, the remainder of the chapter focuses on the operation and utilization of applications that provide a diagnostic testing capability.

The DNS

This section examines the Domain Name System (DNS), and the database contained on a series of servers that make up the DNS. In doing so, it examines the purpose of the DNS, its structure, and the type of records stored on a DNS server.

Purpose

The purpose of the DNS is to provide the TCP/IP community with a mechanism to translate host addresses into IP addresses because all routing is based on

an examination of IP addresses. To accomplish this translation process, a series of Domain Name Servers are used to create a distributed database that contains the names and addresses of all reachable hosts on a TCP/IP network. That network can be a corporate intranet, the portion of the Internet operated by an Internet service provider (ISP), or the entire Internet.

The Domain Name Structure

Internet host names employ a hierarchical address structure. This address structure consists of a top-level domain, a sub-domain, and a host name.

Initially, top-level domain names such as .com, .gov and .edu, as well as IP addresses, were assigned and maintained by the Internet Assigned Numbers Authority (IANA), which was responsible for the overall coordination and management of the DNS. Controversy about the IANA having sole control of top-level domains occurred during the past few years, with the result that the Internet Corporation for Assigned Names and Numbers (ICANN) was formed as a nonprofit organization to take over responsibility for the allocation of IP address space as well as for DNS and root server management. The prior controversy resulted because DNS management and IP address allocation occurs on a global basis, while most of those functions were previously performed under U.S. Government contract by IANA and were not globally representative. Today, ICANN is responsible for the top-level domains and the management of root servers that operate at the top of each defined domain. In comparison, domain administrators where a domain can be assigned to a government agency, university, or commercial enterprise are responsible for host names and IP address assignments within their domains.

The Domain Name Tree

Exhibit 6.1 illustrates a portion of the domain name tree, with the top-level domains consisting of either three-letter top-level domains or two-letter top-level domains. The two-letter top-level domains represent country domains, such as Australia (.au), France (.fr), Israel (.il), etc. There are currently seven top-level three-letter domains as indicated in Exhibit 6.1. In comparison, there are over 100 two-letter country identifier domains.

When an organization applies for an IP address and domain name, both entries are added to the appropriate server at the domain root. For example, if an organization is assigned the domain *widgets.com* as a commercial organization, an entry indicating the network address for *widgets.com* and the domain widgets would be placed in the root .com domain name server.

In examining the entry under the .com domain in Exhibit 6.1, one notes the sub-domain labeled “widgets.” Under the widgets entry, there are two entries: ftp and www. Here, ftp and www represent two host names within the widget sub-domain. The fully qualified names of each host then become ftp.widgets.com and www.widgets.com. Thus, if some one does not know

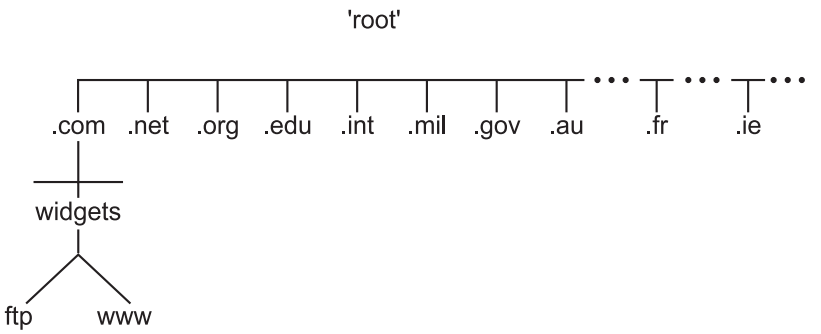


Exhibit 6.1 The Domain Name Tree

the IP address of the ftp and the Web server operated by widgets.com, they can enter the fully qualified domain name for each server, and DNS will automatically perform the translation, assuming applicable DNS entries exist in a server. Thus, one can now look at how host names are converted into IP addresses, a process referred to as name resolution.

The Name Resolution Process

An IP network must have either a local DNS or employ the facilities of another organization's domain name server. For either situation, when you enter a fully qualified host name in a TCP/IP application, the application looks up the IP address of the DNS previously configured for the protocol stack to use. The local computer then transmits an address resolution request using UDP on port 53 to the IP address of the DNS. That IP address could be a DNS on the local network or the DNS operated by the organization's ISP.

Upon receipt of the address resolution request, the DNS first checks its cache memory in an attempt to determine if the IP address was previously resolved. If so, it responds to the computer's request with the host's associated IP address, allowing the computer to use the destination host IP address to create an IP datagram that a router can route. If the DNS did not previously learn the IP address and is not responsible for the domain where the fully qualified domain name host resides, it will forward the request to a higher level in the DNS hierarchy. To do so requires the DNS to have a pointer record that literally points to the address of the next-level DNS. For example, a DNS on a local network would have a pointer record to the DNS operated by the Internet service provider (ISP) that provides the organization with access to the Internet. If the ISP's DNS does not have an entry for the requested host, another pointer record will be used to route the address resolution request to a "higher authority." That higher authority could be a Network service provider (NSP) and eventually the top-level DNS for the domain of the fully qualified host name.

Data Flow

To illustrate the potential flow of data during the address resolution process, consider Exhibit 6.2. In Exhibit 6.2, the user at host gil.smart.edu just entered the host name www.cash.gov into his or her browser and pressed the Enter key, which in effect commences the resolution process. When the address resolution process commences, a UDP datagram flows to the local DNS on the domain smart.edu as indicated by ①. Assuming that the DNS does not have an entry for the network address of the requested host (www.cash.gov), the resolution request flows upward to the next DNS via the use of a pointer record in the local DNS. This is indicated by numbers ②, ③, and ④ in Exhibit 6.2. Assuming the next DNS, which is shown as serving the domain isp.com, does not have an entry for www.cash.gov, the resolution request continues its flow up the DNS hierarchy until it will either reach a server that can resolve the request or arrives at the top-level DNS for the domain for which the host name is to be resolved. This is indicated by ⑤, ⑥, and ⑦ in Exhibit 6.2.

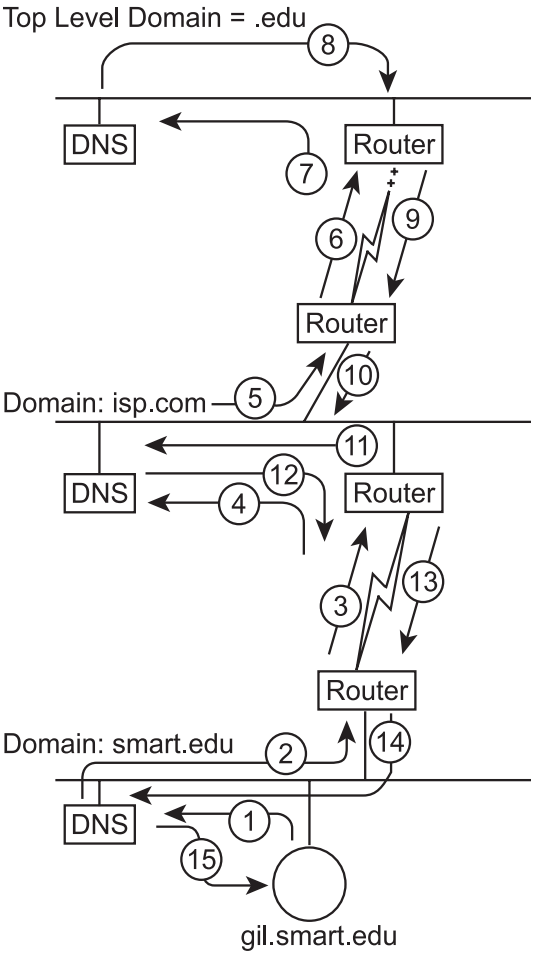


Exhibit 6.2 Potential Dataflow during the Address Resolution Process

Once the address is resolved, the resolution does not flow directly back to the original DNS. Instead, the resolution flows back to each DNS in the hierarchy, providing each server with the ability to update its resolution table. This is indicated by ⑨, ⑩, ⑪, ⑫, ⑬, and ⑭, in Exhibit 6.2. Finally, the local DNS returns the resolved IP address, as indicated in ⑮. At this point in time, the station can now form an IP datagram using a destination IP address obtained from the address resolution process.

Time Consideration

If a fully qualified domain name cannot have its IP address resolved by the local DNS, one or more additional servers must be queried. This means that datagrams conveying address resolution information will flow over relatively low-speed WAN connections for which the time delay then depends on the operating rate of those connections and other activity flowing on each connection, as well as the processing being performed by routers that form the WAN. Because the DNS resolution process on a host results in the setting of a timer, if too much time occurs during the resolution process, the timer will timeout or expire. When this situation occurs, an error will be generated by the protocol stack that will be used by the application to generate an error message. One popular error message generated by a browser informs the user to "check the destination name spelling and try again!" The reason this message does not mention anything about the address resolution process is probably due to the fact that most people using browsers have no knowledge of the process and a more descriptive error message might be counterproductive.

DNS Records

Each DNS can contain a series of different types of records as well as multiple records for one or more record types. Exhibit 6.3 lists some of the more popular types of DNS records.

Exhibit 6.3 Examples of DNS Record Types

Record Type	*Description*
A	Contains an IP address to be associated with a host name
MX	Contains the address of a mail exchange system(s) for the domain
NS	Contains the address of the name server(s) for the domain
CNAME	Canonical Name records contains an alias host name to associate with the host names contained in the record
PTR	Contains a host name to be associated with an IP address in the record
SOA	The Start of Authority records indicate the administrative name server for a domain as well as administrative information about the server

Exhibit 6.4 The File smart.edu.zone

```
;Start of Authority (SOA) record
 smart.edu. IN SOA dns.smart.edu.owner.smart.edu(
                19960105 ;serial#(date format)
                   10800 ;refresh(3 hours)
                      3600 ;retry(1 hour)
                   605800 ;expire(1 week)
                   86400) ;TTL(1 day)
;Name Server (NS) record
 smart.edu. IN NS  dns.smart.edu.

;Mail Exchange (MX) record
 smart.edu. IN MX  20 mail.smart.edu
;Address (A) records.
 router.smart.edu. IN A    198.78.46.1
 dns.smart.edu.    IN A    198.78.46.2
 mail.smart.edu.   IN A    198.78.46.3
 gil.smart.edu.    IN A    198.78.46.30
;Aliases in canonical Name (CNAME) record
 www.smart.edu IN CNAME gil.smart.edu.
```

In examining the record types listed in Exhibit 6.3, note that a domain can have multiple name servers or multiple mail exchange servers. Also note that while the A record provides information necessary for an address resolution process, the PTR record type supports reverse lookups. Exhibit 6.4 illustrates an example of a UNIX Zone file named "smart.edu.zone" for the domain smart.edu. Assume that the Class C address 198.78.46.0 was assigned to the domain smart.edu. Further assume that the server name, dns.smart.edu, is the name server, and that mail.smart.edu is the name of the mail server.

In examining the entries in Exhibit 6.4, note that the string "IN" is used to indicate an Internet address and dates from a period where different types of addresses could be placed in a DNS database. Also note that names and host addresses end with a trailing dot (.) or period to indicate that they are an absolute name or address rather than a relative address.

The SOA Record

The first record normally placed in a Zone file for a domain server is the Start of Authority (SOA) record. Not only does this record govern the manner by which a domain name server and secondary servers, if any, operate, but in addition the ability to read the contents of this record can provide information about the manner by which another domain operates. As noted later in this chapter, one can examine the contents of a domain name server database through the use of the NSLOOKUP application program.

The serial number in the SOA record identifies the version of the DNS database. This value can be used by secondary servers as a metric concerning updating as the number increments whenever the database changes. The refresh value informs the server how often to check for updated information. If the secondary server cannot connect to the primary, it will use the retry value as the time period to wait before retrying. The expire time tells the secondary server when to stop answering queries about the primary when it cannot contact the primary. This value assumes that no answer is better than a bad answer and is shown set to a week (604,800 seconds) in Exhibit 6.4.

Checking Records

Upon further examination of the entries in Exhibit 6.4, one notes that the router in the 198.78.46.0 network has the host address .1, while the DNS has the host address .2, and the mail server has the address .3. Also note that the host gil.smart.edu has the alias www.smart.edu, and that the entry of either host name will return the IP address 198.78.46.30. Thus, by checking the records in a name server, it becomes possible to not only obtain the IP address for a particularly qualified domain name, but, in addition, to discover the alias or aliases assigned to one or more hosts in a domain. Given an appreciation for the role and operation of the domain name system and the servers used in the DNS, one can now focus on use of a series of built-in diagnostic tools provided as application programs in most versions of TCP/IP.

Diagnostic tools

Most operating systems with a TCP/IP protocol stack include several application programs that can be used to obtain information about the state of the network or a particular host. Examples of such applications include Ping, traceroute, nslookup, and finger. Each of these applications will be covered in this section.

Ping

Based on contradictory tales, the name "ping" was given to an application because it either resembled the use of radar or functioned as an anacronym for the full name, Packet Internetwork Groper. Regardless of whether the function of electronic equipment or the development of an anacronym accounted for its name, Ping is one of the most widely used tools — if not *the* most widely used tool — bundled as an application in TCP/IP software.

Operation

Through the use of the Ping application program, a series of Internet Control Message Protocol (ICMP) Echo type messages are transmitted to a distant host.

If the host is both reachable and active, it will respond to each ICMP Echo message with an ICMP Echo Response message. Not only does the use of Ping then tell you that the distant host is both reachable and active, the application also notes the time the echo left the computer and the reply was received to compute the round-trip delay time. Because timing can be very critical for such applications as Voice over IP and interactive query/response, the use of Ping may inform one ahead of time whether or not an application is suitable for use on the Internet or a corporate intranet.

Implementation

There is no standard that governs the manner by which Ping is implemented and different vendor versions, such as UNIX and Windows NT, may differ slightly from one another. One common form of the Ping command to invoke this application is shown below:

```
ping [-q l-v] [-r] [-c Count] [-I Wait] [-s size] host
```

where:

`-q` selects quiet mode that only results in the display of summary information at start-up and completion

`-v` selects verbose output mode that results in display of ICMP packets received in addition to Echo Requests

`-r` selects a route option that displays the route of returned datagrams

`-c` specifies the number of Echo Requests to be sent prior to concluding the test

`-i` specifies the number of seconds to wait between transmitted datagrams containing an Echo Request

`-s` specifies the number of data bytes to be transmitted

`host` specifies the IP address or host name of the destination to be queried

In examining the above options, note that some older implementations of Ping would run until interrupted with a CTRL-C unless a count value was specified through the use of the `-c` option. Also note that many versions of Ping differ with respect to the default wait time between transmitted Echo Requests. Some implementations may transmit echo requests 250 ms apart as a default, while other implementations may use a default of 500 ms, one second, or some other time value. A third item concerning the options listed above concerns the packet size specification variable, `-s`. This variable is used to specify the number of data bytes transmitted and results in a total packet size becoming the specified packet size plus 8, because there are eight bytes in the ICMP header. This means that the default on some implementations is 56 bytes, which results in a 64-byte packet. Given an appreciation for the options supported by Ping, one can now focus on its use within a TCP/IP environment by examining the use of the Microsoft Windows version of Ping, which one can access from the command prompt in Windows.

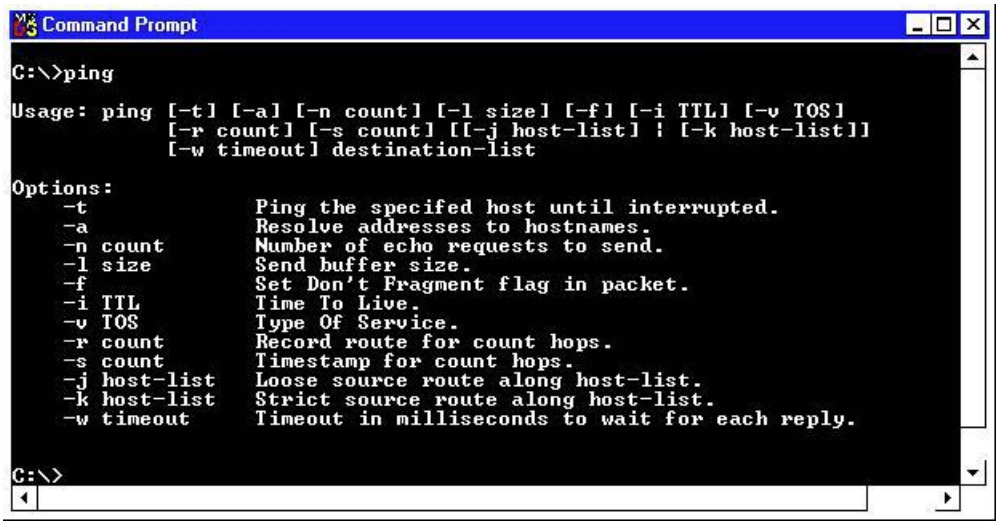

Command Prompt
C:\>ping
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
[-r count] [-s count] [[-j host-list] | [-k host-list]]
[-w timeout] destination-list
Options:
-t Ping the specifed host until interrupted.
-a Resolve addresses to hostnames.
-n count Number of echo requests to send.
-l size Send buffer size.
-f Set Don't Fragment flag in packet.
-i TTL Time To Live.
-v TOS Type Of Service.
-r count Record route for count hops.
-s count Timestamp for count hops.
-j host-list Loose source route along host-list.
-k host-list Strict source route along host-list.
-w timeout Timeout in milliseconds to wait for each reply.
C:\>

Exhibit 6.5 Microsoft Windows Ping Options

Using Windows NT Ping

Exhibit 6.5 illustrates the Windows NT Ping help menu that is displayed when one enters the name of the application without options. In examining the help screen shown in Exhibit 6.5, note that the `-t` option will result in the Ping application continuously transmitting Echo Request packets until interrupted. Unfortunately, this is a favorite attack method used by unsophisticated hackers. Its use is discusssed later in this chapter and in more detail in Chapter 8 when discussing security in more detail. Also note that Microsoft supports several route options as well as a time to live (TTL) option. Typically, most applications set a TTL default value of 250 to prevent a datagram from infinitely wandering the Internet or a private intranet. As the datagram is received by a router, it decrements the TTL value by 1 and compares the result to zero. If the value is greater than zero, it forwards the datagram; otherwise, it places the datagram into the great bit bucket in the sky. By setting the TTL value higher than the default, one can then obtain the capability to reach a host that requires routing through a large number of routers that might otherwise be unreachable from a particular location.

To illustrate the use of Ping, one can ping two locations on the Internet. The first location is the real Whitehouse Web site located at www.whitehouse.gov. The top portion of Exhibit 6.6 illustrates this operation. Note the response "Request timed out" displayed four times. Microsoft's implementation of Ping results in four Echo Request ICMP packets being transmitted as IP datagrams to the destination specified in the Ping command line. The reason the request timed out has nothing to do with the TTL value. Instead, the Whitehouse uses a firewall to block pings because it is one of a number of weapons unsophisticated hackers like to use. Chapter 8 provides greater detail concerning how one can block pings.

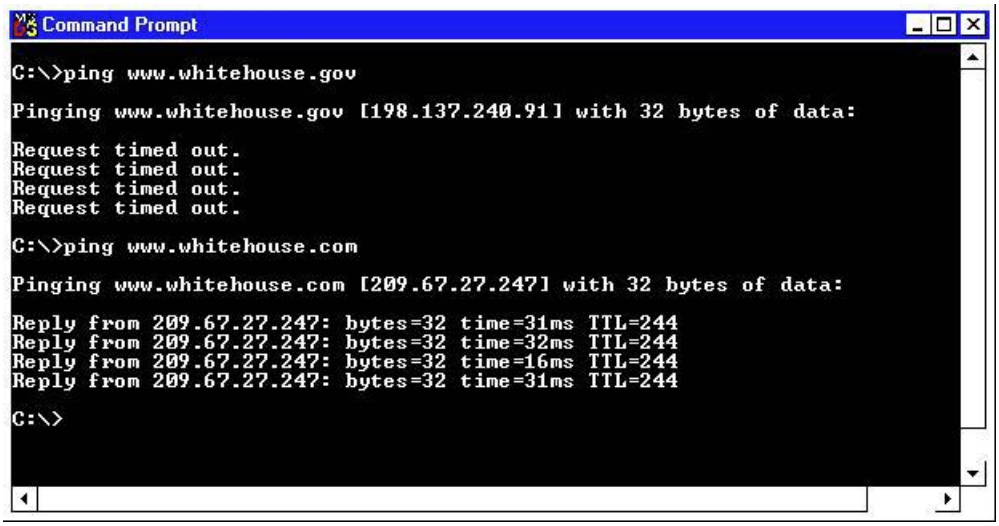

Command Prompt
C:\>ping www.whitehouse.gov
Pinging www.whitehouse.gov [198.137.240.91] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
C:\>ping www.whitehouse.com
Pinging www.whitehouse.com [209.67.27.247] with 32 bytes of data:
Reply from 209.67.27.247: bytes=32 time=31ms TTL=244
Reply from 209.67.27.247: bytes=32 time=32ms TTL=244
Reply from 209.67.27.247: bytes=32 time=16ms TTL=244
Reply from 209.67.27.247: bytes=32 time=31ms TTL=244
C:\>

Exhibit 6.6 Using Ping

The lower portion of Exhibit 6.6 illustrates a second ping, a commercial site Web server whose address is similar, but not the same as "the White House." This commercial site Web address is www.whitehouse.com. Note that Ping automatically resolves the entered host name into an IP address. Also note from the four replies that the round-trip delay varied from a low of 16 ms to a high of 32 ms. This variance is due to the fact that the path between source and destination is subject to random dataflows from other users. This can delay the datagrams one's host is transmitting that contain ICMP Echo Requests.

Resolution Time Considerations

One item that deserves a bit of attention is the fact that it is quite possible for the first response to be much longer than subsequent responses. The reason for this is the fact that if one enters a host name that was not previously resolved, DNS will be required to obtain the IP address associated with the name entered on the command line. Although the example shown in Exhibit 6.6 does not indicate a long delay and, in fact, the first response was 1 ms less than the second, this is not always the case. If a site that is not that popular and whose IP address and host name was not previously learned, one might require information about round-trip delay. To consider a time-dependent application, such as Voice over IP, it is a good idea to periodically transmit pings throughout the day and discard the first response if it appears high due to the address resolution process.

Applications

Although Ping is quite often used to determine round-trip delay, that is not its primary use. Whenever a station is configured and connected to a network,

one of the first things one should do is ping the station. If one obtains a response, this will indicate that the TCP/IP protocol stack is active. This will also mean that the station is properly cabled to the network and that its network adapter is operational. Otherwise, the protocol stack, cable, or network adapter may represent a problem. One can check out the protocol stack by pinging the address 127.0.0.1 or any address on the 127.0.0.0 network because this invokes a loopback. If one obtains a valid result, one would then run diagnostics on the network adapter card provided by the vendor and check or swap cables with a device known to work to isolate the problem. If one attempts to ping a host on a different network, it may not be a simple process to walk over to the destination if all one receives is a time-out message. The cause of a lack of response can range in scope from an inoperative router to an inactive destination. Fortunately, one can obtain insight into the route to the destination through the use of another program called traceroute.

Traceroute

Traceroute, as its name implies, traces the route to a specified destination that will be placed in the application command line. Similar to Ping, there are several variations concerning the implementation of traceroute. A common form of the traceroute command on a UNIX host is:

```
traceroute [-t count] [-q count] [-w count] [-p portnumber]
host
```

where:

- `-t` specifies the maximum time-to-live (TTL) value, with a default of 30 used
- `-q` specifies the number of UDP packets transmitted with each TTL setting; the default is usually 3
- `-w` specifies the time in seconds to wait for an answer from a router
- `-q` represents an invalid port address at the destination; port 33434 is commonly used

Operation

A better understand traceroute options requires an explanation of the manner by which this application operates. Thus, prior to observing the operation of the program and discussing its options, one can focus on how the program operates.

Traceroute works by transmitting a sequence of UDP datagrams to an invalid port address on the destination host. Using common default settings, traceroute begins by transmitting three datagrams, each with its TTL field value set to 1. As soon as the first router in the path to the destination receives the datagram, it subtracts 1 from the value of its TTL field and compares the result to 0. Because the value equals 0, the datagram will be considered to have

expired, and the router will return an ICMP Time Exceeded Message (TEM) to the originator, indicating the datagram expired. Because the originator noted the time the datagram was transmitted and the time a response was received, it is able to compute the round-trip delay to the first router. It will also note that the IP address of that router is contained in the datagram transmitting the ICMP TEM message.

To locate the second router in the path to the destination, traceroute will increment the TTL field value by 1. Thus, the next sequence of datagrams will flow through the first router, but will be discarded by the second router, resulting in another sequence of TEM messages being returned to the originator. This process will continue until the datagrams either reach the destination or the default TTL value is reached, and the application operating on the source terminates. If the datagrams reach the destination, because they are attempting to access an invalid port on the destination host, the destination will return a sequence of ICMP destination unreachable messages, indicating to the traceroute program that its job is finished. With an understanding of how the program operates, one can examine its use with a version included in Microsoft's Windows operating system.

Using Microsoft Windows Tracert

The Microsoft Windows version of traceroute is named tracert. This application program is similar to Ping in that it is operated from the command prompt within Windows.

Exhibit 6.7 illustrates the use of the tracert program without any parameters to display a help screen for the program. In examining Exhibit 6.7, note that

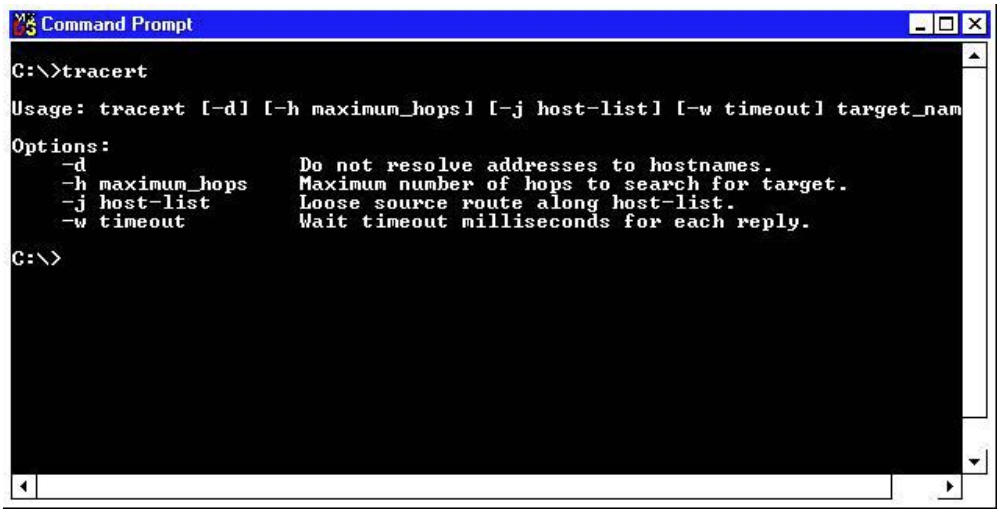

Command Prompt
C:\>tracert
Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_nam
Options:
-d Do not resolve addresses to hostnames.
-h maximum_hops Maximum number of hops to search for target.
-j host-list Loose source route along host-list.
-w timeout Wait timeout milliseconds for each reply.
C:\>

Exhibit 6.7 Microsoft's Implementation of Traceroute (Called tracert) Provides Users with Four Options

the Microsoft implementation of traceroute supports four options. Probably the most commonly used option is the `-h` option, the use of which allows one to change the TTL default of a maximum of 30 hops normally used by the program.

Tracing a Route

To illustrate how tracert can supplement the use of Ping, one can utilize the former to trace the route from the author's network to the real Whitehouse, the one operated by the federal government. In the previous attempt at pinging the Whitehouse, efforts were not successful because each ping returned a timeout message.

Exhibit 6.8 illustrates the use of Microsoft's version of traceroute to trace the route to the Whitehouse Web server. Note that when the program is first executed, it performs an address resolution and displays the IP address of the destination. Also note that the program displays the fact that it is tracing the route to the destination using a maximum of 30 hops, which represents the default value of the application.

Note from Exhibit 6.8 that there were eight routers in the path to the Whitehouse, after which one could not access the Whitehouse network. The eighth router was located in Herndon, Virginia, and according to information returned by the router is operated by PSI.net, an Internet service provider. It was not possible to trace the full route into the Whitehouse network because the router at the Whitehouse Web site was programmed to block both pings and traceroutes. Thus, this resulted in the generation of a "destination net unreachable" message.

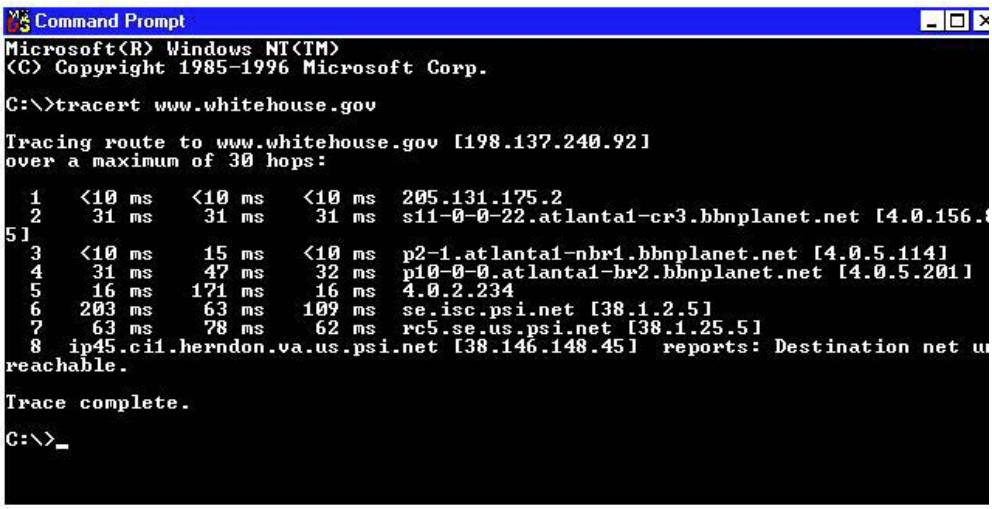

Command Prompt
Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.
C:\>tracert www.whitehouse.gov
Tracing route to www.whitehouse.gov [198.137.240.92]
over a maximum of 30 hops:
1 <10 ms <10 ms <10 ms 205.131.175.2
2 31 ms 31 ms 31 ms s11-0-0-22.atlanta1-cr3.bbnplanet.net [4.0.156.
5]
3 <10 ms 15 ms <10 ms p2-1.atlanta1-nbr1.bbnplanet.net [4.0.5.114]
4 31 ms 47 ms 32 ms p10-0-0.atlanta1-br2.bbnplanet.net [4.0.5.201]
5 16 ms 171 ms 16 ms 4.0.2.234
6 203 ms 63 ms 109 ms se.isc.psi.net [38.1.2.5]
7 63 ms 78 ms 62 ms rc5.se.us.psi.net [38.1.25.5]
8 ip45.ci1.herndon.va.us.psi.net [38.146.148.45] reports: Destination net u
reachable.
Trace complete.
C:\>

Exhibit 6.8 Using Microsoft's Tracert to Trace the Route to the Whitehouse Web Server

In examining the entries in Exhibit 6.8, one also sees that the Microsoft implementation tries three times or more to accurately transmit a sequence of three datagrams with the same TTL field values. Focus now on the round-trip delay and router for each route. The first path, which is from the author's workstation to the router located at IP address 205.131.175.2, required under 10 ms for each of three datagrams to reach, and for the computer issuing the tracert to receive a response. The second path was to the router operated by bbn.planet in Atlanta and resulted in a round-trip delay of 31 ms from the author's computer to that router. Looking at the router information returned, one sees that some routers provide a description of their location and operator and other identifiers, while other routers simply provide their IP address. While all routers in this example returned some information, upon occasion some routers may not respond to a TTL field value of zero condition and will simply throw the datagram away. When this situation occurs, the traceroute program's attempt will timeout and information for that router hop will be denoted through the use of an asterisk (*) as being unavailable.

Applications

As indicated by this particular use of traceroute, this utility program traces the route to a destination. In doing so, it displays the round-trip delay to each router hop, enabling one to determine if one or more routers are causing an excessive amount of delay on the path to a destination. Many times, traceroute can be a valuable tool in determining where network bottlenecks reside. In addition, one can use this tool as a mechanism to identify, to a degree, where along a path a failure of a communications circuit or hardware occurred if a destination should become unreachable. The reason for "to a degree" is due to the fact that if either a circuit becomes inoperative or a router fails, traceroute would not be able to distinguish between the two situations. Before traceroute can be used to isolate the general location of a problem, it is a valuable tool one should consider using either by itself or as a supplement to Ping.

NSLOOKUP

A third built-in application program that can be used to provide valuable information is NSLOOKUP. Unlike Ping and traceroute, which are implemented in essentially all versions of TCP/IP software, NSLOOKUP is available in most, but not all, operating systems that support TCP/IP.

Operation

NSLOOKUP is a name server lookup program. This program can be employed to examine entries in the DNS database of a particular host or domain. There are several ways NSLOOKUP can be implemented, with the most common being an interactive query mode. In the interactive query mode, one would

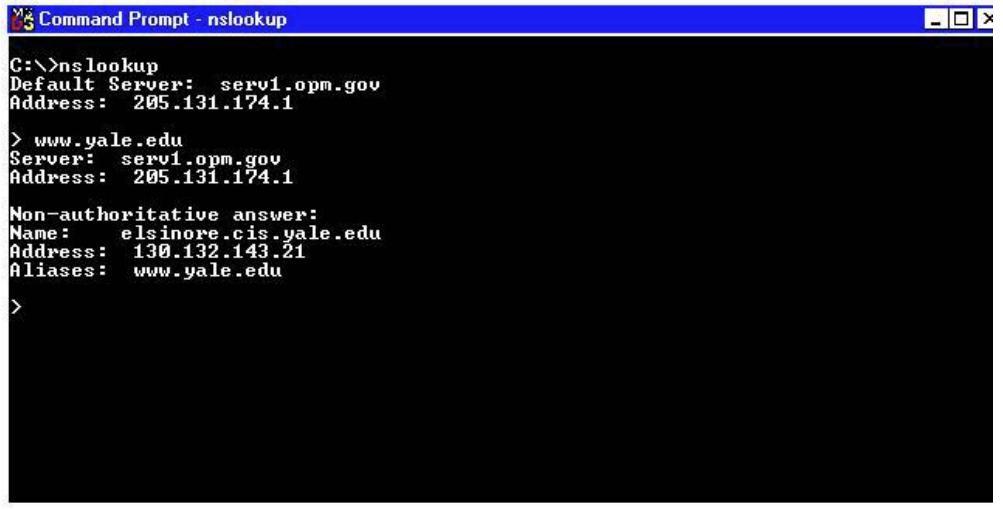


Exhibit 6.9 Using Microsoft's NSLOOKUP to Query the Yale University Server

simply type the command "nslookup." The other method NSLOOKUP supports is a single query mode. The general format of the latter is:

```
nslookup [IP-address\host-name]
```

If one enters the program name by itself, one will be placed in its interactive mode. In the interactive mode, the program uses the greater than sign (>) as a prompt for input. Exhibit 6.9 illustrates an example of the use of NSLOOKUP. In this example, after entering the command "nslookup," the program responds with the name and address of the default name server. This is the name server whose address is configured in the TCP/IP protocol stack operating on the workstation one is using to run the program. That name, server, which is serv1.opm.gov., in this example will be used to resolve each request.

In the example shown in Exhibit 6.9, the next step is to enter the Web server host address for Yale University. Note that NSLOOKUP not only resolved the IP address of www.yale.edu, but, in addition, provided the true name of the Web server because the response indicated that www.yale.edu is an alias.

In the lower portion of Exhibit 6.9, note the prompt in the form of a greater than sign (>). Because the interactive query mode of NSLOOKUP was used, this prompt indicates that it is waiting for an NSLOOKUP command. Because NSLOOKUP queries a name server, one can use the program to retrieve information about different types of name server records. To do so, one must use the "set type=" command, followed by the record type, and then inform one's local DNS server of the distant DNS to be queried.

Exhibit 6.10 provides a list of NSLOOKUP set of query record types one can enter to display a particular type of domain name server record. For example, entering "set q=VID" would be used to specify a query based on user ID.

Exhibit 6.10 NSLOOKUP Set Querytype Values

Nslookup: set q[uerytype]

Changes the type of information query. More information about types can be found in Request For Comment (RFC) 1035. (The set type command is a synonym for set querytype.)
set q[uerytype]=value
Default = A.

Parameters

Value	
A.	Computer's IP address
ANY	All types of data
CNAME	Canonical name for an alias
GID	Group identifier of a group name
HINFO	Computer's CPU and operating system type
MB	Mailbox domain name
MG	Mail group member
MINFO	Mailbox or mail list information
MR	Mail rename domain name
MX	Mail exchanger
NS	DNS name server for the named zone
PTR	Computer name if the query is an IP address, otherwise the pointer to other information
SOA	DNS domain's start-of-authority record
TXT	Text information
UID	User ID
UINFO	User information
WKS	Well-known service description

Finding Information about Mail Servers at Yale

Exhibit 6.11 represents a continuation of the querying of the Yale University DNS. In this example, the record type was set to MX and the domain yale.edu entered. This resulted in the local DNS springing into action and returning a sequence of information about the mail server used at Yale. In examining the entries in Exhibit 6.11, one sees that the response to the query resulted in a listing of both mail exchanger and name server host addresses and IP addresses for that university, providing significant information about its network resources.

Viewing the SOA Record

One can continue the quest for knowledge about Yale University by changing the record type to SOA and again entering "yale.com" as the domain name. Exhibit 6.12 illustrates the resulting display from the previously described

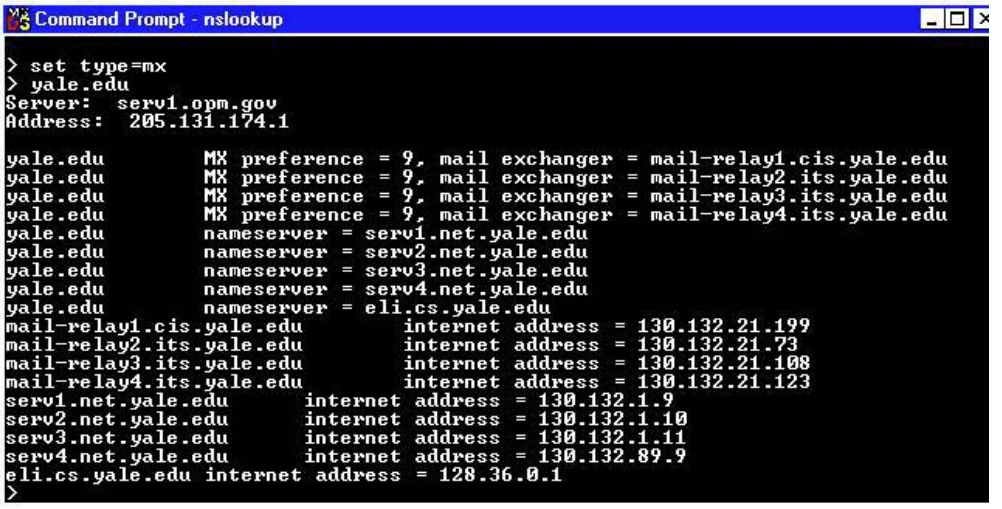

Command Prompt - nslookup
> set type=mx
> yale.edu
Server: serv1.opm.gov
Address: 205.131.174.1
yale.edu MX preference = 9, mail exchanger = mail-relay1.cis.yale.edu
yale.edu MX preference = 9, mail exchanger = mail-relay2.its.yale.edu
yale.edu MX preference = 9, mail exchanger = mail-relay3.its.yale.edu
yale.edu MX preference = 9, mail exchanger = mail-relay4.its.yale.edu
yale.edu nameserver = serv1.net.yale.edu
yale.edu nameserver = serv2.net.yale.edu
yale.edu nameserver = serv3.net.yale.edu
yale.edu nameserver = serv4.net.yale.edu
yale.edu nameserver = eli.cs.yale.edu
mail-relay1.cis.yale.edu internet address = 130.132.21.199
mail-relay2.its.yale.edu internet address = 130.132.21.73
mail-relay3.its.yale.edu internet address = 130.132.21.108
mail-relay4.its.yale.edu internet address = 130.132.21.123
serv1.net.yale.edu internet address = 130.132.1.9
serv2.net.yale.edu internet address = 130.132.1.10
serv3.net.yale.edu internet address = 130.132.1.11
serv4.net.yale.edu internet address = 130.132.89.9
eli.cs.yale.edu internet address = 128.36.0.1
>

Exhibit 6.11 Using NSLOOKUP to Retrieve MX Records from the Yale University Name Server

operations. In examining the entries in Exhibit 6.12, note that Yale University operates four name servers. Also note that the IP address for each server has also been obtained.

Protecting Server Information

One common method of hacker attack is to obtain information about one or more users by listing A records. Due to this, many organizations will block

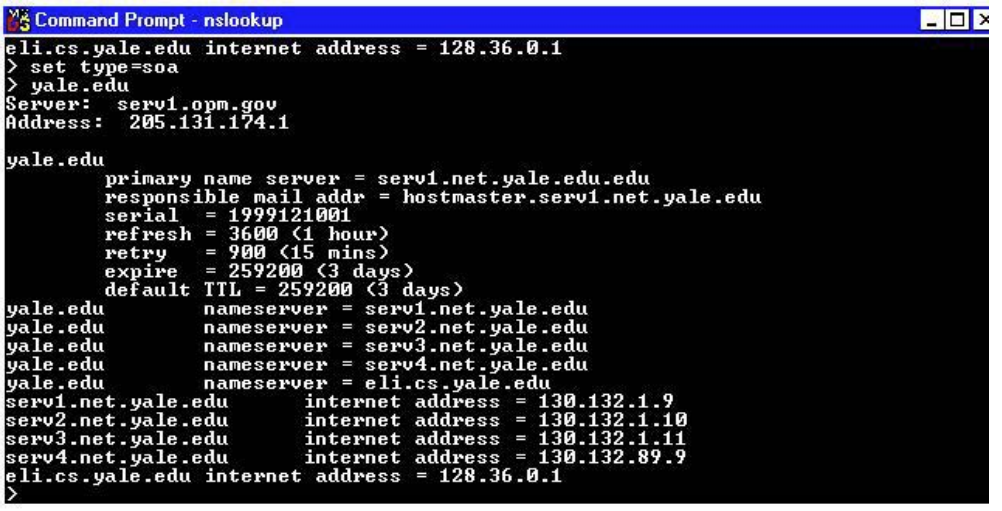

Command Prompt - nslookup
eli.cs.yale.edu internet address = 128.36.0.1
> set type=soa
> yale.edu
Server: serv1.opm.gov
Address: 205.131.174.1
yale.edu
primary name server = serv1.net.yale.edu.edu
responsible mail addr = hostmaster.serv1.net.yale.edu
serial = 1999121001
refresh = 3600 (1 hour)
retry = 900 (15 mins)
expire = 259200 (3 days)
default TTL = 259200 (3 days)
yale.edu nameserver = serv1.net.yale.edu
yale.edu nameserver = serv2.net.yale.edu
yale.edu nameserver = serv3.net.yale.edu
yale.edu nameserver = serv4.net.yale.edu
yale.edu nameserver = eli.cs.yale.edu
serv1.net.yale.edu internet address = 130.132.1.9
serv2.net.yale.edu internet address = 130.132.1.10
serv3.net.yale.edu internet address = 130.132.1.11
serv4.net.yale.edu internet address = 130.132.89.9
eli.cs.yale.edu internet address = 128.36.0.1
>

Exhibit 6.12 Reading the Start of Authority (SOA) Records at Yale University through the Use of NSLOOKUP

the ability of those records to be retrieved. Thus, if one sets the record type to "A" and again enters the domain yale.com, one would not obtain a listing of A records because Yale blocks their retrieval by foreign name servers.

Finger

Finger, a fourth built-in utility, is a program that enables a user to obtain information about who is logged onto a distant computer or to determine information abut a specific user. The use of this command results in a new verb referred to as "fingering," which is not a rude gesture, but a query on the Internet.

Format

The general format of the finger command on a UNIX system is:

```
finger [username] @ {host.name\IP.address}
```

Exhibit 6.13 illustrates the finger command options under Microsoft Windows operation system. Note that the `-l` option results in a long display that can provide detailed information about a user or host computer.

Security Considerations

Similar to other network utility programs under the Microsoft operating system, finger runs in the command prompt dialog box as a DOS application. Because

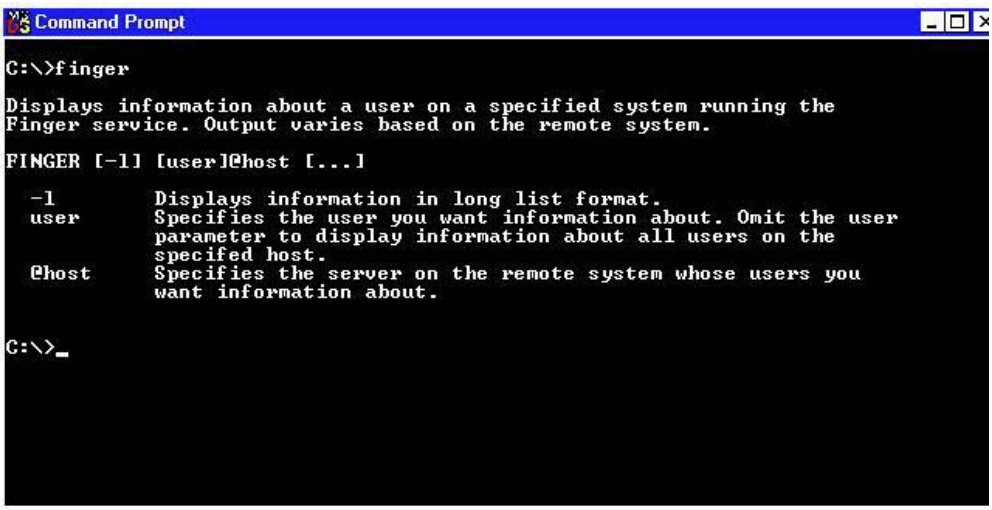

Command Prompt
C:\>finger
Displays information about a user on a specified system running the
Finger service. Output varies based on the remote system.
FINGER [-l] [user]@host [...]
-l Displays information in long list format.
user Specifies the user you want information about. Omit the user
parameter to display information about all users on the
specifed host.
@host Specifies the server on the remote system whose users you
want information about.
C:\>

Exhibit 6.13 The Finger Help Screen under Microsoft Windows

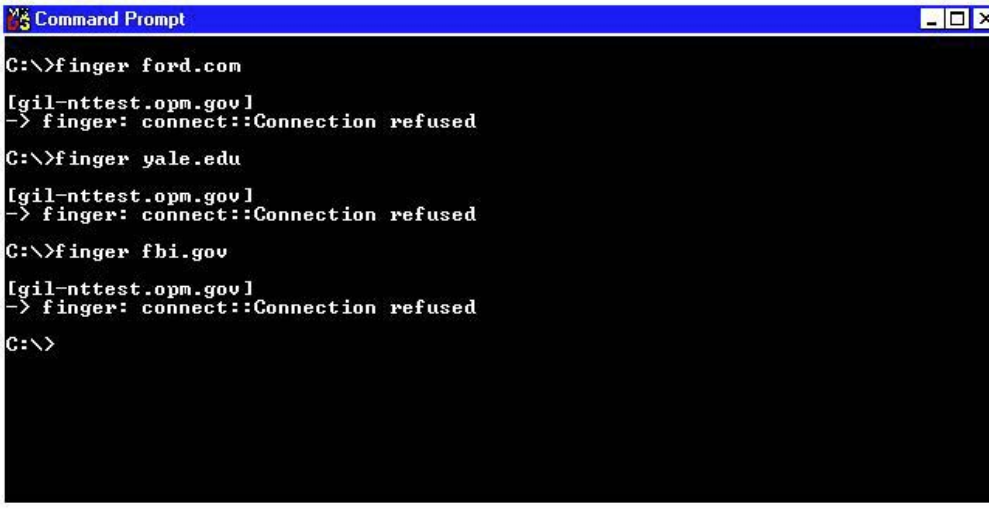


Exhibit 6.14 Many Organizations Will Block Fingering as a Security Measure

the use of finger can provide detailed information about a user or host, it is normally blocked by programming a router to bar datagrams that contain the destination port that identifies a finger application. An example of finger blocking is shown in Exhibit 6.14. In this illustration, the author attempted to finger several domains. First, this author fingered ford.com without success. Next, a U.S. government agency; followed by an attempt to finger Yale University; and, finally, the Federal Bureau of Investigation. Each of these finger attempts was unsuccessful as those organizations block fingering as a security measure.

Applications

As indicated in Exhibit 6.14, many organizations block fingering as a security measure. Thus, a logical question is, "Why discuss its use?" The reason is that many organizations will operate fingering internally, but block its flow into the network. Then, people within an organization obtain the ability to query a host or user to determine who is working on the host, their telephone number, the application they are using, and other information that may be of assistance when attempting to solve a problem.

As indicated in this chapter, the TCP/IP protocol suite contains several built-in application programs that can be used to determine information about hosts, the paths between networks, and users on a host. By carefully considering the use of different application programs, one can obtain valuable tools that will assist in ensuring that if problems occur, one can focus attention on the potential location and perhaps even the cause of the problem.

Chapter 7

Routing and Routing Protocol

Having read the preceding chapters in this book, one is now aware that routing on a TCP/IP network occurs based on the IP address contained in a datagram. One is also aware of the fact that when entering a host address into an application program that address must be translated into an IP address because routing occurs based on the destination IP address and not on the host name. Chapter 6 discussed how the address translation process occurs and the role of the domain name system (DNS) and the entries in the domain name servers that form the DNS. What has not been discussed heretofore is the manner by which a router learns where to forward a datagram based on its destination IP address. Thus, the focus of this chapter is on routing and routing protocols that enable datagrams to flow over a TCP/IP network or between separate networks so that they can reach their destination.

Both routing and routing protocols represent complex topics for which many books have been written. Because the focus of this book is on obtaining a firm understanding of how the TCP/IP protocol suite operates, the focus here is on routing concepts and methods instead of the minute details associated with numerous routing protocols. Doing so will provide the reader with an appreciation for the manner by which datagrams are routed instead of obtaining information required by some people to specifically tailor equipment for operating with a certain routing protocol.

Recognizing the importance of the Internet, this chapter first examines how this mother of all networks is subdivided into separate entities and how the entities are interconnected to one another. In doing so, one obtains an overview of the basic utilization of several types of routing protocols that are responsible for developing paths between networks. Because routers construct routing tables and periodically advertise the contents of such tables to other

routers, this topic is also examined. For those of us from Missouri, the "show me" state, this chapter concludes with an examination of how two popular routing protocols operate in order to obtain an appreciation of the manner by which both routers in the Internet and on private TCP/IP networks understand where to route datagrams.

Network Routing

For a large network such as the Internet or a private network operated by a multinational corporation, it would more than likely be impractical for each router to have entries for each network address. Even if memory was free, whenever a table update was broadcast to adjacent routers, the time required to transmit routing table entries could become so long that it would preclude the ability to transport production data for significant periods of time. Recognizing this potential problem, the various committees responsible for the development of the TCP/IP protocol suite also developed a series of routing protocols. Some protocols are used to convey information within a network consisting of two or more subnetworks managed by a common entity, with the collection of networks referred to as an autonomous system. Other protocols are designed to convey information between autonomous systems. Thus, rather than one routing protocol, the TCP/IP protocol suite supports a family of routing protocols. Because routing methods within an autonomous system differ from routing protocols used to interconnect autonomous systems, one can view the Internet or a corporate enterprise network as a global network and examine the manner by which routing occurs within a global system.

Routing in a Global System

Exhibit 7.1 illustrates an example of a global system consisting of several interconnected autonomous systems. To facilitate reference to protocols, addressing is indicated in terms of two decimal numbers separated by a decimal point instead of true dotted decimal notation.

Autonomous Systems

In examining Exhibit 7.1, one can first more narrowly define an autonomous system. As previously mentioned, it represents a collection of networks managed by a common entity. In actuality, it is the routing protocol that is managed, with the result that only a single routing protocol is used within an autonomous system. Thus, one can also view an autonomous system as a group of networks that use routers to exchange routing information between subnetworks in the system via the use of a common routing protocol.

Each network shown in Exhibit 7.1 can represent a corporate network, educational network, or governmental network. When connected to the Internet through the services of an Internet service provider (ISP), the ISP represents

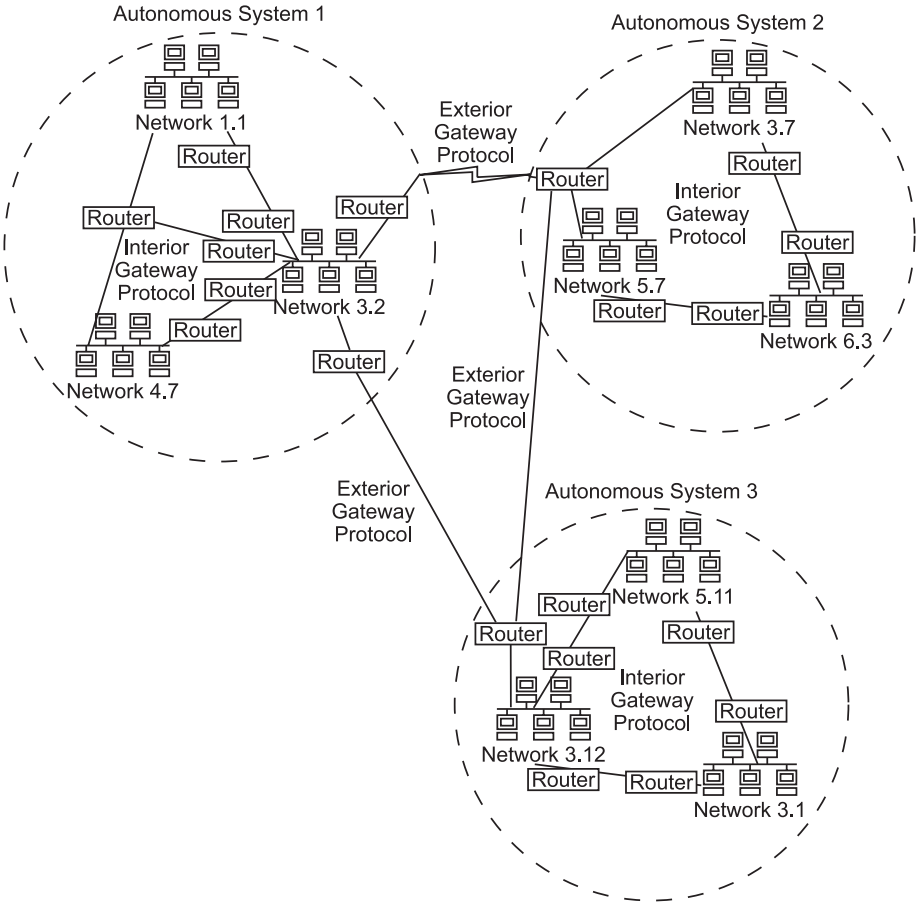


Exhibit 7.1 A Global System Using Different Types of Protocols to Advertise Reachable Information

an autonomous system. If Exhibit 7.1 represents a private enterprise network, perhaps autonomous system 1 represents North America, system 2 represents South America, etc. Thus, each subnetwork in an autonomous network could represent a series of LANs and routers that connect offices in California and the Pacific Northwest, Texas, the southwestern United States, etc. Because organizations acquire IP addresses at different points in time, there is no structure associated with an address relationship between networks in an autonomous system. This explains why the individual networks in autonomous system 1 are numbered 1.1, 3.2, and 4.7 in the example, while the networks located in autonomous system 2 are numbered 3.7, 5.7, and 6.3. For example, in real life, an ISP in Chicago might be responsible for providing routing and connectivity information for a mixture of Class A, B, and C networks whose IP addresses span the gamut of the valid range of addresses available for each class. The only restriction concerning addressing is the fact that each address must be within the allowable range; no single network address can be repeated anywhere else in the global system.

Types of Routing Protocols

There are two general types or categories of routing protocols that provide routing information within and between autonomous systems. Those routing protocols are referred to as Interior Gateway Protocols (IGPs) and Exterior Gateway Protocols (EGPs). They derive their middle identifier due to the fact that when the Internet was developed, the device that provided routing between networks was referred to as a gateway. In some trade literature, the terms "Interior Router Protocol" (IRP) and "Exterior Router Protocol" (ERP) are now used. Although these latter terms are more representative of the devices used in networks to transmit routing information, this author likes both aged wine and long-used terms. Thus, the use of the term "gateway" when describing routing protocols continues to be used herein.

- *Interior Gateway Protocol.* The function of an Interior Gateway Protocol (IGP) is to transmit routing information between routers within an autonomous system. Because all routers in the autonomous system that provide interconnectivity between networks are controlled by the governing authority of the system, it becomes possible to enhance both efficiency and compatibility by specifying the use of a common routing protocol. This routing protocol is used for interconnecting separate networks within the autonomous system. Thus, routing protocols used within each network may or may not be the same as the routing protocol used to interconnect networks within the autonomous system. In fact, it is entirely possible that one or more networks in the autonomous system use "bridging" rather than routing to govern the flow of data within the network.
- *Exterior Gateway Protocol.* Because the routing method used within one autonomous system can differ from that used by another system, it is important to have a mechanism that transfers a minimum level of information between systems. Thus, the purpose of an Exterior Gateway Protocol (EGP) is to transport routing information between routers that connect one such system to another.
- *IGP versus EGP.* Because there are numerous paths between networks in an autonomous system and while one or a few connections link such systems together, there are considerable differences between an IGP and an EGP. For example, an IGP needs to construct a detailed model of the interconnection of routers within an autonomous system. This model must have a sufficient database to compute both an optimum path to a destination as well as obtain knowledge of an alternate path or paths if the least costly path should become inoperative. In comparison, an EGP only requires the exchange of summary information between networks. For example, within autonomous system 1, each router that connects one network to another must know the possible paths from one network to another as well as the other networks in the system. In comparison, to interconnect autonomous systems, an EGP mainly has to convey information between routers linking each system concerning the networks reachable on each system.

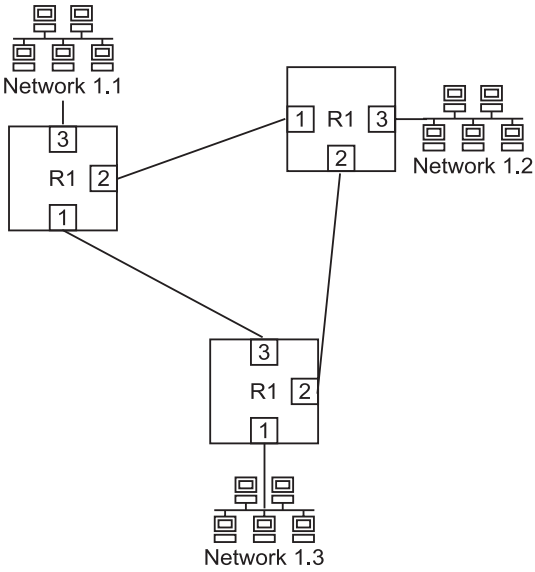


Exhibit 7.2 A Three-Network Autonomous System

Recognizing the differences between an IGP and an EGP, one can now focus on the general type of information included in routing tables and the rationale for routers having to advertise the contents of their routing tables to neighbors.

Need for Routing Tables

In a small system, it is both possible and relatively easy to configure a router so that all paths to other networks are known. To illustrate this, consider Exhibit 7.2, which shows three interconnected networks labeled 1.1, 1.2, and 1.3. The three networks are interconnected through the use of three routers labeled R1, R2, and R3, with each router having three ports labeled 1, 2, and 3.

To facilitate the routing of datagrams, a router must know how to transmit data to another network. In actuality, a router only needs to know what port to output a datagram to reach a given network. Thus, it is entirely possible to configure a static routing table for router R1 shown in Exhibit 7.2 as follows:

Port	*Network*
1	1.3
1	1.3
2	1.2
3	1.1

In the preceding example, the term "static" is used to signify that the entries are permanent and do not vary. While there may be the need for dynamic

routing tables, in many situations, static routing remains a practical solution for configuring routers. For example, if an organization uses one router to connect a LAN to the Internet via an ISP, it makes sense and enhances router performance to use static routing. This is because the organization's router only needs to know the address of the ISP's router. By using static routing in this situation, the organization's router avoids transmitting router table updates, enabling less bandwidth required for overhead and more bandwidth becoming available for actual data transfer.

Returning to the previous example, a problem with the above configuration is the fact that it does not indicate alternate paths between networks, For example, if the circuit between router R1 and router R2 failed, the above configuration does not indicate that datagrams could flow to network 1.2 via router R3.

If one wanted to reconfigure router R1 with knowledge of all possible paths to the three networks, one possible port-network table would be as follows:

Port	*Network*
1	1.3
1	1.2
2	1.2
2	1.3
3	1.1

In examining the preceding port/network table, note that there is no mechanism to distinguish the fact that routing a datagram via a particular port number to a network results in either direct or indirect routing. For example, from router R1 the transfer of a datagram via port 1 provides a direct route to network 1.3. If the datagram is transmitted via port 2, the datagram will have to be relayed via router R2 to reach network 1.3. Thus, another metric is required to distinguish direct paths from indirect paths. That metric is a hop count, which indicates the number of routers a datagram must flow through to reach a particular network. Thus, the routing table for router R1 might be revised as follows:

Port	*Network*	*Hop Count*
1	1.3	1
1	1.2	2
2	1.2	1
2	1.3	2
3	1.1	0

In examining the preceding port/network/hop count table, note that a direct connection to a network results in a router hop count of zero. Also

note that the preceding table provides the ability to distinguish the best route from one that requires more hops. What happens if a path between routers becomes inoperative? For example, consider the path between routers R1 and R2. If the circuit for this path becomes inoperative, how does router R1 obtain information to update its routing table? This update should allow the router to note whether or not a path is available or not available for use. Thus, to dynamically change routing, a router needs to know the state of paths between networks. To obtain this information, a router periodically transmits information to other routers. This information not only tells one router that the network is reachable via another network, but in addition, the lack of an update within a predefined period of time could be used to inform the other router that the path between routers is not available for use. Then the other router will search its routing table, and if another route to a destination is available, make that the available route. Because timing is critical, routers also timestamp information stored in their routing tables. Depending on the manner by which a particular routing protocol is implemented, the timestamp may simply be used to purge entries from a routing table, provide a mechanism for selecting one entry over another, or perform another function.

Routing Table Update Methods

There are two methods routers use to provide other routers with information concerning the contents of their routing tables. One method is for the router to periodically broadcast the contents of its routing table to other routers. This table update method is used by vector distance routing protocols. Here, the term "vector distance" relates to the type of information transmitted by the router and conveyed by the protocol. The vector identifies a network destination, while the distance represents the distance in hops from the router to a particular network destination.

As networks grew in size, the number of routers also increased. As this situation occurred, the number of entries in each router's routing table considerably expanded. At a certain point, the number of routing table entries can result in the periodic transmission of table entries between routers consuming too much network bandwidth. Thus, this potential development led to the creation of a second type of routing table update. This type of routing table update occurs by transmitting routing information only when there is a change in one of its links. The protocol that transmits the changes is referred to as a link state protocol. In addition to providing more efficient utilization of bandwidth, a link state protocol provides the ability to use multiple paths to a common destination. Unlike a vector state protocol that only supports one route at a time, a link state protocol can support load balancing on multiple paths. This is not without a price, as a link state protocol is typically more complex than a vector state protocol. One popular example of a vector distance protocol is the Routing Information Protocol (RIP). An example of a link state protocol is the open Shortest Path First (SPF) protocol.

The Routing Information Protocol

As previously mentioned, the Routing Information Protocol (RIP) represents one of the most popular vector distance routing protocols. Under RIP, participants can be classified as being either active or passive. Active participants can be considered to represent routers that transmit the contents of their routing tables. In comparison, passive devices listen and update their routing tables based on information provided by active routers. Normally, host computers operate as passive participants in a network, while routers operate as active participants.

Illustrative Network

To illustrate the operation of RIP requires the presence of a network. Because an RIP database maintains information about the link between networks, each link in the network is numbered. In addition, because each router represents a node in a network, for simplicity of illustration, the contents of routing tables are shown in terms of their connected links and number of hops required to reach other nodes in a network. Thus, Exhibit 7.3, which will form the basis for examining how RIP operates, shows a four-node network with five numbered links.

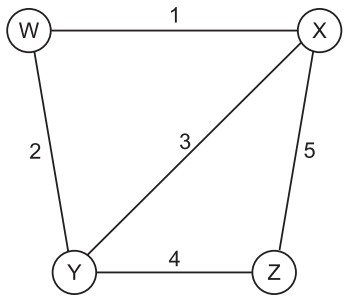


Exhibit 7.3 A Network of Four Nodes Using Five Links

Dynamic Table Updates

Under RIP when a router is "powered on," it only has knowledge of its local condition. Thus, upon initialization, each router will construct a routing table that contains a single entry. For example, the table for router n, where n can represent any router in a network upon power-up would have the following table entry:

From n to	*Link*	*Hop Count*
N	Local	0

Thus, for the router represented by node W in Exhibit 7.3, its routing table would be as follows upon initialization:

From W to	*Link*	*Hop Count*
W	Local	0

Under RIP, an active router will broadcast the contents of its routing table every 30 seconds. Thus, at t = 30 seconds after initialization, node W will broadcast its distance vector (W = 0) to all of its neighbors. Using Exhibit 7.3 to illustrate the operation of RIP, this means that because nodes X and Z are neighbors of node W, they will each receive the distance vector transmitted by node W.

Node C receives the distance vector W = 0 on link 1. Upon receipt of this message, node X updates its routing table by adding 1 to the distant vector supplied by node W. Thus, distance vector table for node X would appear as follows:

From X to	*Link*	*Hop Count*
X	Local	0
W	1	1

At this point in time, node X had a change in its distance vector table. Thus, when 30 seconds expire since its last table update transmission, it transmits a new distance vector set of information. This information would inform adjacent nodes on links 1, 3, and 5 that X = 0 and W = 1.

As node X's routing table update information flows to node Y, it now becomes possible for that node to be aware of node W although there is no direct path between nodes W and Y. Because the distance vectors from node X inform node Y of the hop count from X to itself and to W, node Y adds 1 to each hop count and stores the information in its routing table. If one logically assumes that node Y was powered on, it already had an initial entry to itself in its routing table. Thus, upon receipt of the two distance vector items of information from node X, node Y's routing table will have three entries. Those entries would be:

From Y to	*Link*	*Hop Count*
Y	Local	0
X	5	1
W	5	2

Note that in the preceding routing table, node Y now knows it can reach node W via link 5 and that it requires two hops to reach that node. Thus, although nodes W and Y are not directly connected to one another as table information is transmitted to adjacent nodes under RIP, it becomes possible for nonadjacent nodes to discover how to reach one another.

While node Y updates its routing table, it is safe to assume that because at least 30 seconds have transpired, node W's distance vector information reached node Z. Thus, node Z would have updated its routing table as follows:

From Z to	*Link*	*Hop Count*
Z	Local	0
W	2	1

Because the discussion of the update of node Y followed the update of node X, at least 60 seconds transpired, which enables each node to transmit two routing table updates. Thus, node Y would also receive node Z's next routing table update that would be transmitted to adjacent nodes on links 2, 3, and 4. The routing table for node X would then be updated to the following state:

From X to	*Link*	*Hop Count*
X	Local	0
W	1	1
Z	3	1
W	3	2

Note that at this point in time, node X knows two ways to reach node W: via link 1 with a hop count of 1, which represents a direct connection; or via link 2 with a hop count of 2, which represents an indirect connection. Thus, it is possible for RIP to provide a mechanism for routers to develop a routing table that contains alternate paths.

Because node Z transmits its distance vector information on links 2, 3, and 4, that information also flows to both nodes W and Y in addition to X, whose routing table was just updated. Thus, one can also update the previously updated node Y routing table to ascertain the effect of a routing table update received from node Z. Node Y's routing table would now appear as follows:

From Y to	Link	Hop Count
Y	Local	0
X	5	1
Z	4	2
W	4	2
W	5	2

Because alternate routing entry information would grow exponentially as a mesh network grows in size, RIP does not normally store information about duplicate paths to the same node. Instead, when it computes its routing table update and adds 1 to a received hop count for a node, it compares the new value to the existing value if an entry for the node already exists in memory. If the computed value equals or exceeds the existing hop count, the information about the node received via a router table update is discarded. The exception to this situation is if a router is configured to maintain alternate routing entries to use in the event of a link failure.

Basic Limitations

The preceding example provides a general overview of the manner by which RIP enables nodes to learn the topology of a network. Although RIP does not normally provide alternate path information, the periodic transmission of table entries allows new paths to be learned, because existing information in router tables are time-stamped and an aging process will result in the old path being purged from memory. This process takes time, as table updates occur every 30 seconds. For example, it might take five minutes for one node that is ten hops away from a non-adjacent node to learn that a path changed. A second limitation of RIP is the fact that it is limited to the maximum hop distance it supports. This distance is 16 hops, which means that an alternative protocol must be used for very large networks.

RIP Versions

The original version of RIP developed for use in TCP/IP dates to 1988 when it was adapted by the Internet Activities Board and published as RFC 1058. RIP gained widespread acceptance due to its inclusion as a routing protocol in the Berkeley 4BSD UNIX operating system. In fact, today, both UNIX and Windows NT workstations support RIP in passive mode, allowing such devices to receive and process table updates — although as a passive device, they cannot respond to RIP requests nor broadcast the contents of their tables. Workstations that support RIP do so to avoid having to request information from other routers on a network. Although good in theory, most computers today are configured with a default gateway address for simplicity.

Command	Version	Reserved
Family of Net X		Reserved
Net X Address		
Set to 0		
Set to 0		
Distance to Network X		
Family of Net X + 1		Reserved
Net X + 1 Address		
Set to 0		
Set to 0		
Distance to Network X + 1		
⋮ Up to 25 Entries		

Exhibit 7.4 Routing Information Protocol Version 1 Packet Fields

To obtain an appreciation of the difference between the original version of RIP (now referred to as RIPv1) and its successor (RIPv2), turn attention to the fields with the original RIP packet. Once an appreciation for the use of the fields in that packet has been obtained, one will have the foundation to examine the additional features and capabilities provided by RIPv2.

The Basic RIPv1 Packet

As previously mentioned, to obtain a better appreciation of RIPv1 and Version 2 of the protocol that was standardized in 1994, one should focus on the fields in the RIP packet. Exhibit 7.4 illustrates the fields in the RIP packet. Before discussing those fields, one should note that RIP is transported as a UDP datagram. Thus, the fields shown in Exhibit 7.4 would be prefixed with a UDP header and that header, in turn, would be prefixed with an IP header.

Command Field

The Command field identifies the function of the RIP packet. There are five commands, as described below:

Command	*Description*
1	Request for partial or full routing table information
2	Response containing a routing table
3/4	Turn on (3) or turn off (4) trace mode. This is now obsolete.
5	Sun Microsystems' internal use

Version Field

The Version field identifies the version of RIP. Initially, the value of this field was 1 to indicate RIP version 1.

Family of Net X Field

This field indicates the protocol that controls the routing protocol and is set to 2 for IP. Because Xerox Network Services (XNS) also operated over networks when TCP/IP was evolving, this field was included to allow the same RIP frame to be used to support multiple protocol suites. Thus, while a value of 2 is used for IP, the routing protocol also supports RIP for AppleTalk, Novell's NetWare Internetwork Packet Exchange (IPX), and XNS.

Net X Address Field

When the family of Net X field is set to a value of 2, the Net X address field contains the IP address of the destination network. Under RIP version 1, only the first four bytes of a total of 12 available bytes are used, with the remaining 8-byte set to zero.

If the command field has a value of 1, there is only one entry and the net 1 address (first IP address) is set to a value of zero, which means that the packet is a request for an entire routing table.

Distance to Network X Field

Because RIP is limited to supporting a maximum of 16 hops, this field only supports the integers 1 to 16. An entry of 16 in this field indicates that a network is unreachable. The term "count to infinity" is sometimes used to indicate too many hops for RIP to reach a target.

As indicated in Exhibit 7.4, up to 25 entries containing the IP address of a network and the distance of that network can be included in a RIP packet, with a maximum RIP packet limited to 512 bytes in length.

RIPv1 Limitations

In addition to supporting a maximum hop count of 16 — with only a distance of 15 supported because a count of 16 indicates a destination is unreachable — RIP has several additional disadvantages. Those disadvantages include an inability to differentiate between the bandwidth differences on different links and the fact that broadcasts can become significantly large and consume bandwidth that cannot then be used for data transmission. Another limitation of RIPv1 is the fact that this routing protocol requires a subnet mask to be uniform across an entire network. This is because RIPv1 does not support the ability to contain a subnet mask entry in its routing table. Thus, RIPv1 assumes that the subnet mask is the same for all of its configured ports as the subnet whose value it learns for the network identifier.

RIPv2

Recognizing some of the limitations associated with RIPv1, this routing protocol was modified in 1994, resulting in the development of RIPv2. RIPv2 is back-

Command	Version	Reserved
Address Family Identifier		Route Tag
Net X Address		
Subnet Mask		
Next-Hop IP Address		
Metrix		
Address Family Identifier		Route Tag
Net X + 1 Address		
Subnet Mask		
Next-Hop IP Address		
Metrix		
⋮ Up to 25 Entries		

Exhibit 7.5 RIPv2 Packet Fields

ward-compatible with RIPv1. It adds several important features that enhance its capability. The additional features include a text password authentication capability, the inclusion of subnet masks in its routing tables, and a route tag that provides a mechanism for separating RIP routes from externally learned routes. Exhibit 7.5 illustrates the RIPv2 packet format.

In comparing the fields in RIPv1 to RIPv2 packets, one notes the use of several new fields in RIPv2, as discussed below.

Route Tag Field

The purpose of the Route Tag field is to provide RIPv2 with the ability to advertise routes that were learned externally. For example, assume a router is used to provide an interconnection between autonomous systems. It would then learn routers through the use of an EGP and would use the route tag for denoting or identifying the autonomous system from which those routers were learned.

Next Hop Field

The purpose of the Next Hop field is to provide a router with the ability to learn where the next hop is located for the specified route entry. A value of 0.0.0.0 in this field is used to indicate that the source address of the update should be used for the route. This field is primarily used when there are multiple routers on a single LAN segment that use different IGPs for routing updates to different LANs. If a point-to-point link is used, the next hop can be obtained from the source IP address of the IP header of a datagram. Thus, this field is not very useful for point-to-point links.

In examining the fields in the RIPv2 packet shown in Exhibit 7.5, one might be a bit puzzled as to how this newer version of RIP can support authentication. The trick used to obtain this additional feature is for the RIPv2 header to set

Command	Version	Unused
FFFF		Authentication Type
Password		
Password		
Password		
Password		
Address Family Identifier		Route Tag
Net n + 1 Address		
Subnet Mask		
Next Hop		
Metric		

Exhibit 7.6 RIPv2 Authentication Packet

the value of a field within the packet to a special value that tells a receiver to interpret the data differently. One can appreciate technique by examining how RIPv2 supports authentication.

Authentication Support

When the Address Family Identifier field in a RIPv2 packet is set to a value of hex FFFF, the header of the resulting RIP datagram changes into an authentication header. Exhibit 7.6 illustrates the fields in a RIPv2 authentication packet.

In examining Exhibit 7.6, one sees that an Authentication Type field value of 2 is for using a simple password. A field of 16 bytes that allows one to convey up to a 16-character password follows this. If RIPv2 is communicating with a router supporting RIPv1, RIPv1 will ignore this entry because the value of hex FFFF in the fourth field of the header is not recognized as an IP address family.

A RIPv2-compliant router can be configured with or without authentication. If it is configured with authentication disabled, the router will accept and process both RIPv1 and RIPv2 unauthenticated messages. If the router receives a RIPv2-authenticated message, it will discard the message. If the router is configured to support authentication, then unauthenticated messages will be discarded.

Although RIPv2 added additional features, it maintained the maximum distance allowable between two stations at 15 hops. RIPv2 does not consider the fact that there could be other metrics besides a hop count that should be considered when determining a path. These shortcomings were considered in the development of a routing protocol referred to as Open Shortest Path First (OSPF), which is discusssed next.

OSPF

The use of Open Shortest Path First (OSPF) as a routing protocol in place of RIP results in both advantages and disadvantages. Although OSPF is more

efficient in its overall use of bandwidth, it consumes more bandwidth during its initial discovery process and represents a more complex process that consumes more router memory cycles. This chapter section focuses on obtaining an understanding of the manner by which OSPF operates and its key features.

Overview

The Open Shortest Path First (OSPF) routing protocol is a link state protocol that transmits routing table updates either when a change occurs or every 30 minutes via the use of a multicast address. Thus, before considering additional OSPF features, note that its use is not as great a detriment to bandwidth as the use of RIP.

Path Metrics

A second key feature of OSPF is the fact that paths are based on a true metric and not just a hop count. For example, OSPF routers pass messages to each other in the form of Link State Advertisements (LSAs). One type of LSA includes the IP address of a router's interface and the cost of that interface. Here, the cost is configured by the router administrator. While it is possible for a router administrator to associate any value to a router interface, RFC 1253 defines a series of recommendations concerning the assignment of costs to router interfaces for use of OSPF. Exhibit 7.7 illustrates that such costs are relative to a 100-Mbps operating rate.

Initialization Activity

Although the original OSPF routing protocol dates to the days of ARPAnet, it was not until RFC 2178 that the protocol became available for modern TCP/IP networks. Similar to RIP, OSPF is an IGP and is designed to run within a single autonomous system. Upon initialization, each router within an autonomous system records information about each of its interfaces. Each router in the

Exhibit 7.7 Potential OSPF LSA Costs

	Data Rate of Interface	*Cost*
>=	100 Mbps	1
	10 Mbps	10
	E1 (2.048 Mbps)	48
	T1 (1.544 Mbps)	65
	64 Kbps	1562
	56 Kbps	1785
	19.2 Kbps	5208
	9.6 Kbps	10416

autonomous system then constructs a Link State Advertisement (LSA) packet that contains a list of all recently viewed routers and the costs previously associated with their interfaces. Rather than broadcasting the LSAs to all adjacent nodes as supported by RIP, OSPF subdivides a network into geographic entities (known as areas) and forwards LSA packets to routers within its area. A received LSA is then flooded to all other routers in an area, with each router updating its tables with a copy of the most recently received LSA. Thus, each router obtains complete knowledge of the topology of the area to which it was assigned.

Router Types

Under OSPF, a network or a group of networks can represent an area. Through the use of areas, routing table updates can be better controlled, with packet flooding occurring within an area while different areas communicate with one another to obtain information about networks within different areas. This subdivision of labor is based on the use of different types of routers, with the function of routers with respect to OSPF based on their type.

If there is only one area, each router maintains a database of the topology of the area and only one router has to deal with external routes beyond the area. When there are multiple areas, a number of other types of routers may be required to perform specialized operations. In fact, under OSPF, there are six types of routers that can be used. Exhibit 7.8 describes the function of each type of OSPF router.

Message Types

As discussed, OSPF routers transmit messages in the form of LSAs. There are currently six types of LSAs used by the protocol. This chapter section briefly discusses the function of each type of SLA message.

Exhibit 7.8 Types of OSPF Routers

Router Type	*Mnemonic*	*Description*
Backbone router	BR	A router that has a connect to a backbone
Area border router	ABR	A router that has an interface to multiple areas
Autonomous system boundary router	ASBR	A router that exchanges routing information with routers connected to other autonomous systems
Internal router	IR	A router that connects networks within a common area
Designated router	DR	A specified router that transfers information on behalf of adjacent routers on a subnet
Backup designated router	BDR	A router that backs up the designated router and takes over should the primary DR fail

Exhibit 7.9 OSPF Message Types

Type 1	Router Links Advertisement
Type 2	Network Links Advertisement
Type 3	Summary Links Advertisement
Type 4	Autonomous System Boundary Router Summary Link Advertisement
Type 5	Autonomous System External Link Advertisement
Type 6	Multicast Group Membership Link State Advertisement

Type 1 Message

A Type 1 LSA message is used to transmit information about a router's interface and the cost associated with the interface. Because an interface is defined with the use of an IP address, the information in a router's Link State Advertisement consists of an IP address-cost metric pair. Exhibit 7.9 lists the six types of SLA messages OSPF routers use.

Type 2 Message

The purpose of a Type 2 LSA message is to inform all routers within an area of the presence of a designated router (DR). Thus, a DR floods a Type 2 LSA upon its election. This message contains information about all routers in the area, as well as the fact that one is now the DR for the area.

Type 3 Message

Because an area border router (BR) connects adjacent areas, a mechanism is needed to describe networks reachable via the adjacent border router (ABR). Thus, an ABR router floods a Type 3 message into an area to inform routers in the area about other networks that are reachable from outside the area.

Type 4 Message

A Type 4 message describes the cost from the router issuing the message to an autonomous system boundary router. Thus, this message allows a boundary router that functions as a gateway to another autonomous system to note the cost associated with accessing different networks via different paths within its system.

Type 5 Message

A boundary router that connects autonomous systems generates a Type 5 message. This message describes an external network on another system reachable by the router. Thus, this message flows to routers in one autonomous system to describe a network reachable via a different system.

Type 6 Message

The last type of LSA is a Type 6 message. A Type 6 message enables a multicast-enabled OSPF router to distribute multicast group information instead of having to transmit multiple copies of packets.

Operation

The actual initialization of OSPF in an autonomous area requires a series of packets to be exchanged that enables the Designated Router and Backup to be selected and router adjacencies to be noted prior to routing information being exchanged. The most basic exchange between OSPF routers occurs via the transmission of Hello messages. Such messages flow between routers that enable routers within an area to discover one another, as well as to note the relationship between routers. Once the DR and BDR are selected, additional messages are exchanged that eventually enable one area to become aware of other areas, as well as networks reachable outside of the current autonomous system.

Although the initial learning process is complex, once routing table information is constructed, updates only occur when there is a change in the network structure or every 30 minutes. Thus, although OSPF initially requires more bandwidth than RIP, it rapidly reduces its bandwidth requirements.

Chapter 8

Security

One of the key advantages of the TCP/IP protocol suite is also a disadvantage. The advantage is its openness, with RFCs used to identify the manner by which the protocol suite operates. Unfortunately, this openness has a price: it allows just about any person to determine how the various components that make up the protocol suite operate. This capability enables people who wish to do harm to IP networks and computers operating on those networks to develop techniques to do so. In comparison, other protocol suites that were developed by commercial organizations may not be as extensively documented. Thus, people who wish to exploit the weaknesses of a protocol suite developed by a commercial organization may face a far more daunting task.

Because the use of the Internet has expanded at an almost exponential rate, another problem associated with the TCP/IP protocol suite involves the connection of private networks to the Internet. With almost 100 million users now connected to the Internet on a global basis, this means that if only a very small fraction of Internet users attempt to break into different hosts, the total number would become considerable. Recognizing the openness of the TCP/IP protocol suite and the ability of people from Bangladesh to Belize to hack computers, security has become a major consideration of network managers and LAN administrators and is the focus of this chapter.

This chapter focuses on a series of TCP/IP security-related topics. Because the router represents both the entryway into a network as well as the first line of defense, this device is considered first. One can consider using this examining measure to prevent unauthorized entry into a router's configuration sub-system. Also discussed are other methods one can use to create access lists that perform packet filtering. Although the use of router access lists can considerably enhance the security of a network, there are certain functions and features that they do not perform. Thus, many network managers and LAN administrators rightly supplement the security capability of a router with a firewall, the operation of which is also covered in this chapter.

Router Access Considerations

Any network connected to the Internet gains connectivity through the use of a router. This means that if a remote user can gain access into an organization's router, it becomes possible for that user to reprogram the router. This reprogramming could result in the creation of a hole in a previously created access list that enables the person to overcome any barrier the access list created to the flow of packets. If the person is not a simple hacker, but a paid agent, he or she might then reprogram an organization's router to initially transport packets to a location where they are recorded and then forwarded onto their destination. Thus, the ability to control access to a router's configuration capability cannot be overlooked. This represents one of the first facts that one should consider if and when connecting an organization's private network to the Internet.

Router Control

Most routers provide several methods that can be used to control their operation. Although routers produced by different vendors may not support all of the methods to be discussed herein, they will usually support one or more of the methods to be discussed in this section. Those methods include direct cabling via a control port into the router, Telnet access, and Web access.

Direct Cabling

All routers include a control port that enables a terminal device to be directly cabled to a router. The terminal device can be a PC or a dumb asynchronous terminal. Once connected to the router, a user normally must enter a password to access the configuration system of the router. The exception to this is the first time a person accesses the router via a direct connection, because most routers are shipped without a password being enabled. Thus, if someone has unpacked a router, connected a terminal to its direct connect configuration port, and configured an access password, the next person to use the terminal would have to know the password to be able to configure the router.

Benefits and Limitations

The key benefit of a direct cabling configuration method is that only people who can physically access the terminal can configure the router. Thus, this configuration method precludes the ability of a remote user (to include potential hackers) from gaining access to an organization's router configuration substation and changing one or more parameters. Unfortunately, this advantage is also a disadvantage: it precludes the ability to remotely configure a router. Thus, if one's organization has trained personnel at each router location, or if one's budget permits the accumulation of frequent flyer mileage as one

sends employees on the road to reconfigure routers, one can consider the use of direct cabling. If the organization needs to be responsive to changing requirements and cannot afford to have trained employees at each router location, one would then prefer a technique that provides a remote router management capability. This capability can be obtained through the use of Telnet or Web access to the router.

Telnet and Web Access

Some routers include the ability for remote configuration via Telnet. Other routers extend remote router configuration capability to Web browsers via the use of the HyperText Transport Protocol (HTTP). For either situation, all that is required for a remote user to gain access to the configuration subsystem of a router is its IP address and password, assuming a password was previously configured to block casual access into the router's configuration subsystem. An IP address is required because both Telnet and Web access, while possible via a host name resolved into an IP address as well as via a directly entered IP address, is usually not obtainable via the former. This is because many organizations do not assign a host name to a router.

Once a person uses Telnet or a Web browser to connect to a router, that user will then be prompted to enter a password to gain access to the router's subsystem. An example of this situation is illustrated in Exhibit 8.1, which shows the use of the Microsoft Windows Telnet client program in an attempt to gain access to a Cisco Systems router.

In the example shown in Exhibit 8.1, access to a Cisco router can require the use of two passwords. The first password is used to gain access to the router's virtual terminal (vt) port that allows a person to display router information, but precludes the ability to configure the router. If one types the command "enable" to gain access to the router's privileged mode of operation, one will be prompted to enter a second password. When this password is successfully entered, One obtains the ability to both display information as well as to configure the router.

In a Cisco router environment, once one types the command "enable" and enters an applicable password, the system prompt changes from a "greater than" sign (>) to a "pound sign" (#). In the example shown in Exhibit 8.1, the name "Macon" was previously assigned to the router being accessed, resulting in the router name prefixing the prompt that denotes the mode being occupied. Once in the privileged mode, one can change the router's configuration.

Protection Limitation

One of the problems associated with remote access to routers is that the basic method of protection via a password can be overcome if care is not taken during the password creation process. This is because most routers will simply disconnect the Telnet or Web connection after three failed password entry

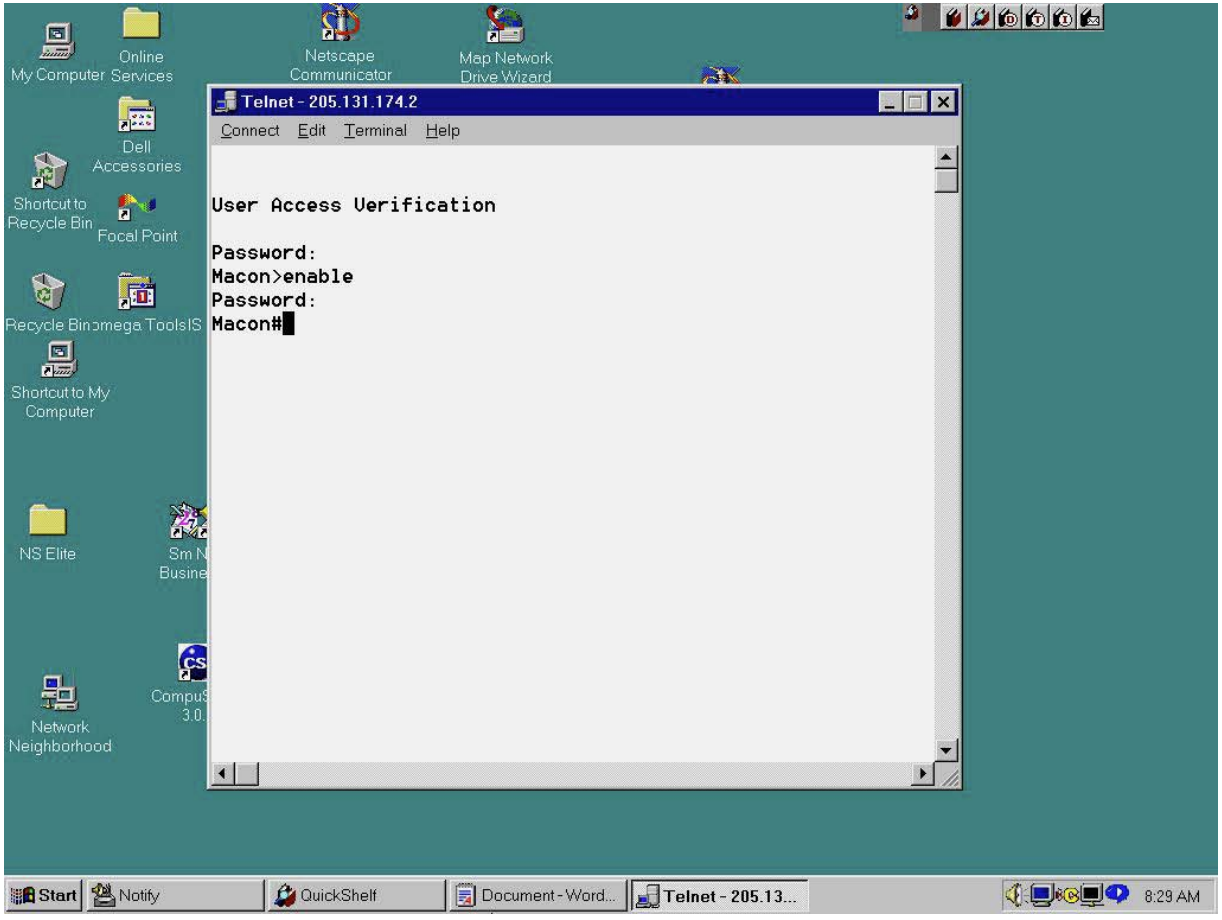


Exhibit 8.1 Using Telnet to Access the Configuration Subsystem on a Router

attempts, enabling a user to try three more passwords. Because of this limitation, several hackers in the past are known to have written scripts that would try three entries at a time from an electronic dictionary. If the network manager or LAN administrator used a name, object, or expression that could be extracted from an electronic dictionary, it was not long before the hacker gained entry into a router's virtual terminal capability and essentially took control of the device. Thus, it is extremely important to use an alphanumeric password combination that is not in a dictionary for password creation. In addition, through the use of an access list, it becomes possible to restrict access to a router's virtual terminal (vt) port from a known IP address. Therefore, it is possible — with some effort — to make it extremely difficult for a remote user other than a designated employee or group of employees to remotely access an organization's routers. There is one limitation associated with the use of a router's access list that warrants discussion. Because an access list would be developed to allow packets from a known IP address to access a router's virtual terminal (vt) port, this means the address cannot be dynamically assigned. In addition to requiring a fixed IP address, the use of an access list to protect access to a vt port also restricts the ability of traveling employees to gain access to a router's vt port. This is because ISPs commonly assign IP addresses dynamically. For example, an employee communication from San Francisco at 3:00 p.m. would have a different temporary IP address than that assigned when he or she accessed the Internet at 2:00 p.m. Similarly, if the employee took a trip to San Jose at 6:00 p.m. and accessed the Internet, another IP address would be temporarily assigned to the employee's notebook computer.

One method to overcome the previously described problem is for traveling employees to dial an access server that is directly connected to the organization's network. If one configures the server to accept a static IP address assigned to an employee's notebook, then a router's access list could be used to allow remote access for authorized employees that travel around the country or around the globe. This action, of course, results in the necessity for employees to use long distance instead of potentially more economical Internet access. Because many organizations only periodically require traveling employees to reconfigure organizational routers, the cost of potential long-distance support may not represent a detriment to its use.

Although the previous discussion focused on Telnet access to a router's vt port, some routers also support configuration via a Web browser that results in similar, but not identical, issues. That is, while one can use a router's access list capability to allow HTTP access from predefined IP addresses, Web browsers also support secure HTTP, referred to by the mnemonic HTTPS. If the router supports HTTPS, then one may be able to support the use of public key encryption and preassigned digital certificates between the Web browser and router, adding an additional level of security to gaining remote access to a router's configuration subsystem.

Given an appreciation for the manner by which local and remote users can gain access to a router's configuration subsystem, one can now focus on

the use of access lists. In doing so, the author examines the rationale for their use, the basic types of access lists, and new capabilities recently added to access lists that considerably enhance their functionality.

Router Access Lists

Because a router is designed to interconnect geographically separated networks, this device also represents an entry point into a network. Recognizing this fact, one of the earliest features added to routers was an access list capability, which is the subject of this section.

An access list, also referred to as an access control list, represents one or more statements that, when executed by a router, perform packet filtering. Access lists can be configured for a variety of network protocols, such as Apple Talk, IP, IPX, DECnet, and Vines. Once an access list is created, it must be applied to an interface to take effect. In doing so, one must consider whether one wants to filter packets flowing into the organization's network from an untrusted area, or packets leaving the organization's network, or both.

Rationale for Use

While the primary use of access lists is to bar traffic to enhance network security, there are other reasons to use this router feature. One reason that is gaining acceptance is the ability to filter packets based upon the IP Type of Service (TOS) field. This allows traffic to be prioritized when entering or leaving a queue. Another reason for using access lists is to implement corporate policy concerning the use of different services on an intranet or accessible via the Internet. For example, corporate policy may be to bar all or certain employees from Web-surfing.

Two additional reasons for configuring access lists include restricting the contents of routing updates and providing a mechanism for traffic flow. While each of these reasons can be sufficient in and of themselves to use access lists, the primary reason for their use is to obtain a basic level of security.

This chapter section primarily focuses on the use of access lists to enhance network security. Because Cisco Systems has a dominant market share of the installed base of routers, examples in this section will be focused on Cisco access lists. However, other router vendors follow a similar methodology by which access lists are created and applied to an interface, either inbound or outbound. While one may have to tailor examples in this section to the access list format supported by other router manufacturers, the basic concepts are applicable to most routers. Thus, the information presented in this section should serve as a guide to the use of a router's access list as the first line of defense of a network, regardless of the manufacturer of the device.

Although access lists were developed to support different protocols, the primary focus here is on the IP. This is because IP is the only protocol supported for use on the Internet, and most organizations that use access lists

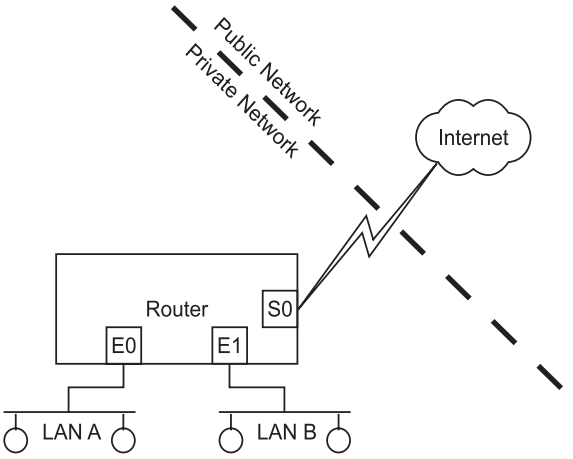


Exhibit 8.2 Connecting a Private Network Consisting of Two LANs to the Internet

for security do so with respect to data flowing to and from the Internet with respect to an organization's private network.

Exhibit 8.2 illustrates the use of a router to connect two LANs — both to each other as well as to the Internet. In this example, the router provides a gateway from the LANs to the Internet. Because many, if not most, organizations do not want to allow any user that accesses the Internet to gain access to their private network, it is quite common for the network manager or LAN administrator to program the router to restrict the flow of packets. This packet flow restriction is accomplished through the use of a router's access list capability, and the technique used by the router in examining packets specified by an access list is referred to as packet filtering.

Ports Govern Data Flow

In examining Exhibit 8.2, note the three router ports labeled S0, E0, and E1. Port S0 represents a serial port that provides connectivity from the router to and from Internet data flows. Ports labeled E0 and E1 represent Ethernet ports that provide connectivity to individual Ethernet LANs that represent an organization's private network.

To use a router to obtain a packet filtering capability requires one to perform two functions. First, one or more access lists must be created. Next, each access list that is created must be applied to a specific port in a specific data flow direction. Data flow direction is defined as follows: data flowing toward a router is considered to flow in, while the direction of data leaving a router is considered to flow out. Thus, in a Cisco router environment, the key words "in" and "out" are used when an access list is applied to an interface to indicate which data flow direction packet filtering specified by the access list should occur.

Data Flow Direction

Because one can apply up to two access lists per router port (one inbound, one outbound), it becomes possible to filter packets at different locations with respect to a router's interfaces. In general, it is a good rule to apply an access list as close as possible to the data source to be filtered. For example, if one wants to filter packets entering the organization's private network from the Internet, one should apply an access list to port S0 in the inbound direction. Although one could create access lists for ports E0 and E1, if one applies the access list to port S0, it would protect the entire network with one access list and avoid a bit of a duplication of effort. This duplication would be necessary if one needed to permit or deny similar data flows to or from each network. For example, if one wanted to block Web surfing from stations on both LANs, one could either code an applicable access list and apply it to interface S0, or code two access lists and apply one to interface E0 and the other to interface E1. Similarly, if one wanted to restrict the ability of network users on LAN A to surf the Web, it would be easier to apply an access list to port E0 in the inbound direction to filter all HTTP data flows than to apply the access list to port S0. If one applied the list to port S0 and simply filtered on HTTP, one could adversely affect the ability of users on LAN B to surf the Internet. This would then require one to program a more complex access list to apply to port S0 to block LAN A users while allowing LAN B users to surf the Web. Thus, for most situations, one should attempt to place an access list as close as possible to the source of data to be filtered.

Given an appreciation for the general use of access list, one can now probe deeper into the specific types of access lists that can be used.

Types of Access Lists

There are two basic types of access lists supported by Cisco router: standard and extended. A standard access list is limited to performing filtering on source addresses. In comparison, an extended access list permits filtering based upon source address, destination address, source port, destination port, and even the setting of bits within certain fields of a packet.

Standard Access Lists

The basic format of a standard IP access list is shown below.

```
access-list list # {permit | deny } source [wildcard-
mask] [log]
```

Although one access list can be assigned to one direction on an interface, a router can support multiple access lists. Thus, the list # represents a decimal from 1 to 99 that identifies a particular access list. In addition, because Cisco

assigns numeric ranges to different types of access lists with respect to the protocols they operate upon, the list # also serves to identify the protocol that will be filtered. This means that a list # from 1 to 99 identifies a standard IP access list.

In examining the above format, note that one would specify either the keyword "permi" or "deny" in an access list statement. The use of "permit" enables a packet to flow through an interface when conditions specified in the access list are matched. Similarly, the use of "deny" sends a packet to the great bit bucket in the sky when conditions specified in the access list are matched.

The "source" entry in the standard access list format represents the IP address of a host or network from which the packet was transmitted. One can specify the source either via the use of a 32-bit IP address denoted in dotted decimal notation, or using the keyword "any" to represent any IP address. The wildcard-mask functions as a reverse-network address subnet mask; that is, one would place 1's in the bit positions to be ignored. For example, assume an organization has the IP class C network address of 198.78.46.0. When configuring the TCP/IP protocol stack, one would use the subnet mask 255.255.255.0 to specify the absence of subnets. Here, the trailing byte of 0's indicates a don't care condition, and results in hosts 1 through 254 being considered as residing on the network. If configuring Cisco access lists, the wildcard mask uses an inverse of the subnet mask, with 1 bits in positions one wants to ignore. Thus, if one wants to allow all hosts on network 198.78.46.0, the wildcard mask would be 0.0.0.255. An example of a standard IP access list statement permitting IP packets from all hosts on the 198.78.46.0 network would be:

```
access-list 1 permit 198.78.46.0 0.0.0.255
```

Note that list number 1 is in the range 1 to 99. Thus, the list number identifies the access list as a standard IP access list. Also note that the keyword "access-list" is entered with a dash (-) between "access" and " list."

Returning to the format of the standard IP access list, note the optional term "log." When used, this option results in an informational logging message about packets that match an access list statement being sent to the console. Logging can be an effective tool for both developing complex access lists as well as for generating information about the number of packets permitted or denied by an access list.

Because it can be tedious to enter wildcard masks for specific network addresses when constructing an access list with a large number of statements, one can use the keyword "host" to reference a specific address. That is, instead of having to enter 198.78.46.8 0.0.0.0 to reference the specific IP address of 198.78.46.8, one could enter "host 198.78.46.8" as a shortcut reference.

To illustrate the use of a standard access list, assume one simply wants to block the ability of users on LAN A in Exhibit 8.2 from accessing the Internet. Further assume that the IP address of the LAN A network is 198.78.460.

Because the access list one creates will be applied to port E0, one must specify that port in an interface command. In addition, one would use the "ip access-group" command to apply an access list to a particular direction. The format of that command is shown below:

```
ip access group [list number] {in/out}
```

Thus, access list would be as follows:

```
interface S0
ip access-group 1 out
access-list 1 deny 198.78.46.0 0.0.0.255
```

Note that the preceding access list blocks all packets from LAN A from flowing into the Internet. The reason one does not use the E0 interface because if one did, it would block LAN A users from accessing LAN B. Also note that when one uses a standard access list, there is no way to specify a particular TCP or UDP application. Thus, one could not use a standard access list to allow FTP, but block HTTP. To obtain an additional packet-filtering capability requires the use of an extended access list.

Extended Access Lists

An extended access list considerably extends the capability of router packet filtering. As previously mentioned, through the use of an extended access list one can filter on source and destination addresses, Layer 4 ports, protocol, and even the bit settings within certain packet fields. The general format of an extended IP access list is as follows:

```
access-list list # {permit|deny} [protocol] source
source- wildcard [port] destination destination -
wildcard [port] [established] [log] [other options]
```

Similar to a standard access list, the list number defines the type of list. To define an extended IP access list, one would use a list number between 100 and 199. Because IP includes ICMP, TCP, and UDP, one can create a more explicit access list statement by specifying a specific IP protocol. In fact, one can even specify a routing protocol in an access list. Note that unlike a standard access list, which is restricted to filtering based on a source address for all IP traffic, one can filter based upon a particular IP protocol, source and destination address, source and destination ports, as well as use such keywords as "established," "log," and other options.

Both source and destination ports are optional and are used when filtering on Layer 4 information. One can specify a port number, the mnemonic of a port number such as SNMP, or use the keyword "RANGE" to create specific ranges of port numbers. One can also use mnemonics to represent numeric

operations, such as GT for "greater than," LT for "less than," and EQ for "equal to." For example, one could specify filtering based on SMTP for source or destination ports or both by substitution, either "EQ 25" or "EQ SMTP" for source port and/or destination port because port number 25 represents SMTP.

The keyword "established" is only applicable for the TCP protocol. When used in an access list statement, a match occurs if a TCP datagram has either its ACK or RST bits sets. This situation occurs when a packet flowing in one direction represents a response to a session initiated in the opposite direction. One common use of an extended IP access list with the keyword "established" is to only permit packets to enter a network from the Internet that represent a response to a session initiated via the trusted private network side of a router. For example, consider the following extended IP access list statement.

```
access-list 101 permit tcp any any established
```

The list number of 101 identifies the access list as an extended IP access list. The protocol specified is TCP and one permits packets to flow from any source to any destination address if their ACK or RST bit is set, which indicates the packet is part of an established conversation. Thus, one would normally want to apply the preceding statement to a serial interface connecting a router to the Internet. However, one would also want to apply the access list containing the preceding statement in the inbound direction if one wants to consider restricting TCP traffic from the Internet to sessions established by hosts on one's internal network or networks connected via the router to the Internet.

To provide the reader with a bit (no pun intended) more information about extended IP access lists, consider the following access list consisting of three statements.

```
access-list 101 permit tcp any any established
access-list 101 permit ip any host 198.78.46.8
access-list 101 permit icmp any any echo-reply
```

The first statement permits TCP datagrams that are part of an established conversation. The second statement permits IP from any host to the specific host whose IP address is 198.78.46.8. Here, the keyword "host" followed by an IP address is equivalent to an IP address followed by a wildcard mask of 0.0.0.0. The third statement in the access list permits ICMP from any host to any host if the packet is a response to a ping request (echo-reply). Note that if one applies this access list in the inbound direction on a serial interface connecting a router to the Internet, the result will be to allow responses to pings originating from the trusted side of the router.

Because an access list is based on the premise that all is denied unless explicitly allowed, if one applies the preceding access list in the inbound direction, pings originating on the Internet destined to all hosts on the 198.78.46.0 network other than host 198.78.46.8 will be blocked. The reason

pings can flow to host 198.78.46.8 is due to the fact that the second statement permits IP traffic to include ICMP to that host. If one wanted to preclude pings to that host, one could insert a deny ICMP statement prior to the permit IP statement. This fact illustrates an important concept concerning access list processing. The contents of packets are matched against statements in an access list in their sequence. If access list statement n permits or denies a packet, then statement n + 1 will not be matched against the packet. Thus, it is important to review the order in which statements are entered into an access list.

To illustrate the additional capability afforded by the use of extended access lists, reconsider the previous problem where all users on LAN A were blocked from accessing the Internet. Now assume one wants to allow employees to Telnet to any location on the Internet, but block Web surfing from users on both LANs. To accomplish this, one would create the following router commands:

```
interface S0
ip access-group 101 out
access-list 101 permit telnet any any
```

Note that it was not necessary to encode a specific access-list deny statement. This is because the end of each access-list has an implicit "deny all" statement that blocks anything that is not explicitly permitted. As indicated by this small example and the prior example in this section, an extended access list significantly extends the capability to perform complex packet filtering operations beyond that supported by standard access lists.

New Capabilities in Access Lists

In tandem with several relatively recent updates to the Cisco Internetwork Operating System (IOS) were improvements to the functionality and capability of access lists. Four additions to access lists include named access lists, reflexive access lists, time-based access lists, and TCP intercept. In actuality, these additions represent additional capabilities added to access lists and not new types of access lists.

Named Access Lists

Because each type of access list has a limited range of acceptable numbers, it is theoretically possible — although highly unlikely — that one could run out of numbers when configuring an enterprise router. Perhaps a more important reason for named access lists is the fact that a name can be more meaningful than a number. In any event, named access lists were introduced in IOS Version 11.2.

As its name implies, a named access list is referenced by a name instead of a number. Revising the previously presented access list into a named extended IP access list called "inbound," one would obtain the following statements:

```
ip access-list extended inbound
     permit tcp any any extended
     permit ip any host 198.78.46.8
     permit icmp any any echo-reply
```

An important aspect of named access lists that deserves mention is the fact that they enable one to delete specific entries in the list. This is accomplished by entering a "no" version of a specific statement contained in a named access list. This action is not possible with a numbered access list. Instead, to revise a numbered access list, one would create a new list, delete the old list, and apply the new list to the appropriate interface.

Reflexive Access Lists

One of the limitations associated with the use of the keyword "established" in an extended IP access list is that its only applicable to TCP. To control other upper layer protocols, such as UDP and ICMP, one would either permit all incoming traffic or define a large number of permissible source/destination host/port addresses. Along with being a very tedious and time-consuming task, the resulting access list could conceivably require more memory than available on the router. Perhaps recognizing this problem, Cisco introduced reflexive access lists in IOS Version 11.3.

A reflexive access list creates a dynamic, temporary opening in an access list. That opening results in the creation of a mirror image or "reflected" entry in an existing access list, hence the name of this list. The opening is triggered when a new IP traffic session is initiated from inside the network to an external network. The temporary opening is always a permit entry and specifies the same protocol as the original outbound packet. The opening also swaps source and destination IP addresses and upper-layer port numbers and will exist until either the session initiated on the trusted network is closed or an idle timeout value is reached.

There are four general tasks associated with the creation of a reflexive access list. First, one would create an extended named access list. In an IP environment, the following command format would be used:

ip access-list extended name

where "name" represents the name of the access list.

Next, one would create one or more permit entries to establish reflected openings. Because a reflexive access list is applied to outbound traffic, it will result in temporary openings appearing in an inbound access list. When defining permit statements for the outbound access list, one would use the following statement format:

permit protocol **any any reflect** name [**timeout** seconds]

One can use the keyword "timeout" to assign a timeout period to each specific reflexive entry created in the inbound direction. If one elects not to use this option, a default timeout of 300 seconds will be used for the opening. One can also elect to place a global timeout on all reflexive statements. To do so, use the following global command:

```
ip reflexive-list timeout value
```

where "value" is the global timeout value in seconds.

The third task is to create an access list for inbound filtering into which dynamic reflexive entries are added. Conclude the operation with the following command:

```
evaluate name
```

where "name" represents the name of the access list and causes packets to be evaluated by reflexive entries.

The following example illustrates the creation of a reflexive access list where the reflected openings are limited to 180 seconds of idle time. In examining the following statements, note that the three deny statements in the extended access list named "inbound" are conventional statements that are not reflected. Also note that those statements are commonly referred to as anti-spoofing entries. That is, many times, hackers use RFC 1918 private network IP addresses in an attempt to preclude network operators from identifying the source address of an attack.

```
!
ip reflexive-list timeout 180
!
ip access-list extended outbound
     permit tcp any any reflect my_session
     permit udp any any reflect my_session
     permit icmp any any reflect my_session
!
ip access-list extended inbound
     deny ip 10.0.0.0 0.255.255.255 any
     deny ip 172.16.0.0 0.31.255.255 any
     deny ip 192.168.0.0 0.0.255.255 any
evaluate my_session
```

Before moving on, it should be noted that while reflexive access lists considerably extend the capability of packet filtering, they are limited to single-channel connections. This means applications such as FTP that use multiple port numbers or channels cannot be supported by reflexive access lists. However, a special release of IOS, initially referred to as the Firewall Feature

Set (FFS), introduced support for dynamic openings in access lists for multi-channel applications. Now referred to as Context Based Access Control (CBAC) in IOS Release 12.0, CBAC also adds Java blocking, denial-of-service prevention and detection, real-time alerts, and audit trails. Unfortunately, CBAC is only applicable for certain platforms.

Time-Based Access Lists

Until IOS Version 12.0, there was no easy method for an administrator to establish different security policies based on the time of day or date. Although an administrator could create multiple access lists and apply them at different times, doing so could be a complex process. In addition, does anyone really want to stay in the office until 6 p.m. on a Friday to implement a new security policy? With the introduction of IOS Version 12.0, time-based access lists provided the flexibility to implement different policies based on time.

In the wonderful world of IP, the use of time-based access lists is a relatively easy, two-step process. First, one would define a time range, and then reference that time range in an access list entry. One can specify a time range using a time-range statement whose format is shown below.

```
time-range time-range-name
```

where "time-range-name" is the name assigned to the time range.

Once the preceding is accomplished, one can specify a time range in one of two ways: use an "absolute" statement or a "periodic" statement, with the format of each shown below:

```
absolute [start time date] [end time date]
periodic days-of-the-week hh:mm to [days-of-the-week]
hh:mm
```

The "time" parameter is entered in the format hh:mm, where hours (hh) are expressed in a 24-hour format. One can list the days of the week separated by spaces or use the keywords "daily" or "weekend." Once the time-range is created, it can be referenced through the optional keyword "time-range" in a traditional access list entry. The following example illustrates the creation of a time-based access list that restricts Web access to Monday through Friday from 8 a.m. to 5 p.m.

```
time-range allow-http
     Periodic weekdays 8:00 to 17:00
!
access-list 101 permit tcp any any eq 80 time-range
         allow-http
```

TCP Intercept

This examination of enhancements to access lists concludes with TCP intercept. This feature was added in IOS Version 11.3 as a mechanism to alleviate a special type of denial-of-service attack referred to as SYN flooding.

Under TCP's three-way handshake, the first packet in a session has the SYN bit sent. The recipient of this initial packet requesting a service, such as HTTP, responds with a packet in which the SYN and ACK bits are set and waits for an ACK from the originator of the session. If the originator of the request fails to respond, the host times-out the connection. However, while the host is waiting for the conclusion of the three-way handshake, the half-open connection consumes resources.

Suppose an unscrupulous person modifies software to transmit tens of thousands of packets with their SYN bit set, using forged IP source addresses. This will result in the attacked host never receiving a response to its request to complete each three-way handshake. Thus, its resources will be consumed as it times-out each session, only to be faced with a new flood of additional bogus packets with their SYN bit set. As host resources are consumed, its ability to be usable decreases to a point where little or no useful work occurs. Because of the popularity of this method of attack, TCP intercept was added as a mechanism to limit half-open connections flowing through a router.

The TCP intercept feature works by intercepting and validating TCP connection requests. The feature operates in one of two modes: intercept or watch. When in intercept mode, the router intercepts inbound TCP synchronization requests and establishes a connection with the client on behalf of the server and with the server on the behalf of the client, in effect functioning as a proxy agent. If both connections are successful, the router will merge them. To prevent router resources from being fully consumed by a SYN attack, the router has aggressive thresholds and will automatically delete connections until the number of half-open connections falls below a particular threshold.

The second mode of operation of TCP intercept is watch mode. Under watch mode, the router passively monitors half-open connections and actively closes connections on the server after a configurable length of time.

Enabling TCP intercept is a two-step process. First, configure either a standard or extended IP access list permitting the destination address one wants to protect. Once this is accomplished, enable TCP intercept using the following statement:

```
ip tcp intercept list list#
```

Because the default mode of operation of TCP intercept is intercept mode, a third step may be required to set the mode. To do so, use the following command:

```
ip tcp intercept mode {intercept | watch}
```

As previously mentioned, TCP intercept includes aggressive thresholds to prevent router resources from being adversely consumed by a SYN attack. There are four thresholds maintained by routers that one can adjust. Those thresholds are set using the following TCP intercept commands, with the default value indicated for each setting:

```
ip tcp intercept max-incomplete low number 90
ip tcp intercept max-incomplete high number 1100
ip tcp intercept one-minute low number 900
ip tcp intercept one-minute high number 1100
```

To illustrate the use of TCP intercept, assume one wants to protect the host 198.78.46.8. To do so while selecting default thresholds would require the following access list statements.

```
ip tcp intercept list 107
access-list 107 permit tcp any host 198.78.46.8
```

Applying a Named Access List

Until now, it has only briefly been mentioned that access lists are applied to an interface. Because some of us are from Missouri, the "show me" state, the following paragraphs review how they are applied by supporting a named access list instead of a numbered one.

As previously mentioned, one applies an access list through the use of the "access-group" command. The general format of that command is:

```
ip access-group {list# | name } {in | out}
```

Note that one would use a name in the access group command to reference a named access list. Also note that one would use either "in" or "out" to reference the direction with respect to the router interface where filtering occurs. Because the access group command only associates the direction of filtering to an access list number or name, one would use an "interface" command before the "access-group" command to tie everything to an interface. For example, assume one wants to apply the extended IP access list named "inbound" previously created to a router's serial port in the inbound direction. The statements would be as follows:

```
interface serial 0
     ip access-group inbound in
!
ip access-list extended inbound
     permit tcp any any established
     permit ip any host 198.78.46.9
     permit icmp any any echo-reply
```

Configuration Principles

Although access lists can be used as the first line of defense to protect a network, there are several configuration principles to note to prevent good intentions from creating protection holes. First, Cisco access lists are evaluated sequentially, top-down. This means that as soon as a match occurs, access list processing against a packet terminates. Thus, it is important to place more specific entries toward the top of an access list. Second, if adding new entries to an access list, they are added to the bottom of the list. This can result in undesirable results and one may wish to consider creating a new list, deleting the old one, and applying the new list instead of attempting to patch an existing list.

Limitations

While router access lists represent an important tool for providing a barrier against unwanted intrusion, they are limited in their capability. For example, if one needs to provide access to a Web server, filtering via a router access list to allow any user on the Internet to only access the server does not block such users from attempting to break into the server. This limitation results from the fact that a router access list does not actually examine the contents of data within a packet. Instead, a router access list examines packet headers for port numbers and IP addresses and enables or disables the flow of information by comparing those metrics against the parameters coded in one or more access list statements. Thus, for many networks, a more powerful security tool in the form of a firewall is both required and added as a network component.

Firewalls

A firewall in some respects is similar to a router in that it is designed to enforce a network access control policy by monitoring all traffic flowing to or from a network. There are many firewall products being marketed, with some devices consisting of software that users load onto a usual LAN port PC, while other products represent a combined hardware and software solution in one package. Regardless of the method by which a firewall is constructed, it is important to note the manner by which it should be installed.

Installation Location

Because a firewall is designed to inspect the contents of packets, it is important to locate the device where it will do the most good. That location is commonly between a public and private network boundary. The term used to denote the location is referred to as a DMZ (for demilitarized) LAN.

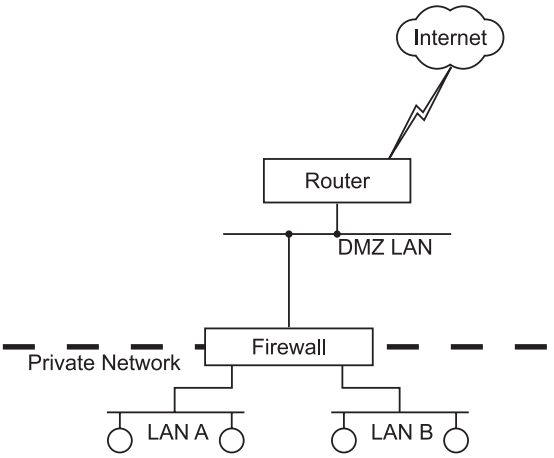


Exhibit 8.3 Locating a Firewall on a DMZ LAN Allows All Packets Flowing from and to the Internet to be Examined

Exhibit 8.3 illustrates the installation of a firewall on a DMZ LAN to protect a private network. Note that the term "DMZ" references a LAN with no attached workstations or servers. Because the only connections to the DMZ are from a router and a firewall, all traffic that flows from the Internet to the private network, and vice versa, must flow through the firewall. Thus, the use of a DMZ in the manner illustrated in Exhibit 8.3 results in all packets flowing from or to the Internet being examined by the firewall.

Basic Functions

Through the ability to "look" into the contents of packets, firewalls become capable of being programmed to examine the contents of information being transferred. This functionality is referred to as "stateful" inspection by one vendor and enables a firewall to look for suspicious activity, such as repeated log-on attempts. Upon determining that a repeated sequence is occurring, such as an attempted log-on, the firewall would either alert the network operator or bar further attempts, with the specific action based on the manner in which the firewall was configured.

When comparing the capability of a firewall to a router, one notes the several functions performed by firewalls that are not commonly associated with routers. Those functions include proxy services, authentication, encryption, and network address translation, as discussed next.

Proxy Services

A proxy can be considered an intermediary that acts on behalf of another. When discussing firewall proxy services, a firewall answers requests on behalf

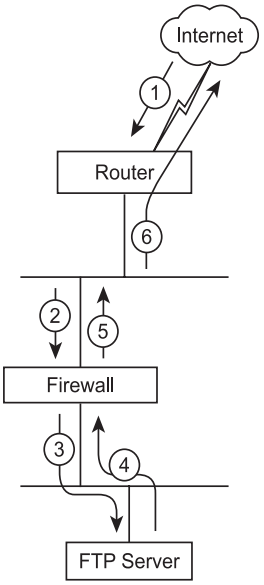


Exhibit 8.4 Dataflow When a Firewall Supports Proxy Services

of another computer, first examining the request to ensure it is acceptable. If found to be so, the firewall will then pass the request to the indicated computer. Rather than responding directly to the original requestor, the destination host responds to the proxy service on the firewall. Exhibit 8.4 illustrates the data flow for a proxy service function operated on a firewall.

There are several applications for which proxy services were developed, such as FTP and HTTP. To understand how proxy services operate and why they are an important security tool, assume an FTP proxy is operational on the firewall shown in Exhibit 8.4. After describing how the proxy operates, one can focus on the flow of data in the example.

FTP includes several commands whose use can be harmful if invoked under certain circumstances. Two of those commands are "MGET" and "MPUT," with the "M" prefix in the "GET" and "PUT" commands used to denote that multiple files will be retrieved from or transferred to an FTP server. To illustrate how these commands might represent a problem, assume an organization has a 56-Kbps connection to the Internet. Further assume that the organization operates a combined FTP/Web server, and FTP provides access to a directory with 10 Gbytes of data.

If a remote user on the Internet accesses the organization's FTP server and enters the command "MGET *.*," this action would cause every file in the directory of the server currently accessed to be downloaded. If there were 10 Gbytes of data in the directory, this one command would result in 396 hours of transmission. This error, in effect, would significantly impact the ability of users to access the organization's Web server and to receive a timely response. Because an FTP application supports all FTP commands, one cannot block MGET and MPUT via an FTP initialization screen. Instead, one must employ a

proxy service on a firewall, as illustrated in Exhibit 8.4. In this example, an FTP request from the Internet (1) flows first to the router whose access list would be programmed to permit inbound FTP to the FTP server. Next, the FTP request would be intercepted and examined by the proxy service on the firewall (2). If the FTP operation was allowed, the firewall would pass the request to the server (3). If not, the firewall would reject the request (5), and the rejection would flow back to the originator via the router (6). Assuming the FTP request was allowed, the firewall forms a new packet to indicate that the request came from that device and records the originator's address plus the source port number in a table in memory prior to forwarding the packet to the server (3). The server will respond to the firewall (4), which will then check its tables and create a new packet, so the original source address is now its destination address. Once this packet reformatting is complete, the firewall transmits the packet to the router (5), which forwards it to the Internet (6) toward its destination.

As indicated by the preceding example, a proxy service on a firewall intercepts packets, allowing the firewall to compare actions within the packet against a predefined configuration that either allows or prohibits such actions. For an FTP proxy, one would more than likely block the use of "MGET" and "MPUT" commands as they could significantly increase network traffic. Although a person could still request one file after another or write a script to execute a series of "PUT" or "GET" commands, this action would be more difficult than simply issuing a single "MGET" or "MPUT" command. In addition, some firewall proxy services permit the user to configure a maximum amount of traffic that can flow to or from a specific IP address, further limiting the use of FTP as a denial-of-service weapon.

Authentication

Although a user-ID/password sequence is commonly considered to represent an authentication method, it provides a limited level of security in comparison to other methods. One common firewall authentication method is obtained through token support. A token represents an algorithm that is used to produce a pseudo-random value that changes every minute. A remote user is then provided with a credit-card sized token generator that displays a five- or six-digit number that changes every minute based on an algorithm embedded in the circuitry on the card. A remote user who needs to be authenticated is blocked from gaining access to services on the network by the firewall. The firewall prompts the user for his or her PIN number and the five- or six-digit authentication number. The firewall uses the PIN number with an algorithm in its software to generate an authentication number that is compared to the transmitted number. If the two match, the user is then authenticated.

The key safety feature of the token lies in the fact that the loss of the card by a remote user does not allow a finder to access the network. A PIN number is required for authentication and, once authenticated, a user must still have the applicable passwords to gain access to hosts on the network. Because of this, a token-based authentication scheme is probably the most popular method for authentication remote users.

Encryption

Another feature added to some firewalls is encryption. Because data flow over the Internet is subject to viewing by numerous organizations that operate routers, encryption is a necessity when using the Internet as a virtual private network (VPN) to interconnect two or more organizational locations. To ensure that organizational information is not read, an encryption firewall will allow the firewall manager to define network addresses for packet encapsulation. Then, packets destined to a different organizational network via the Internet will flow through the Internet in encrypted form, with a newly formed header using the address of a firewall on the destination network because that firewall will now be responsible for decryption. Although the use of VPNs is in its infancy, as their usage increases, one can expect an increase in demand for encryption performing firewalls.

Network Address Translation

In concluding this examination of firewall features, this section focuses on network address translation (NAT). NAT represents both a security mechanism and an address extension mechanism.

The use of NAT results in the translation of an IP address behind a firewall into a new IP address for routing via the Internet. Thus, NAT hides organizational addresses and servers as a mechanism to prevent direct attacks to host on a network.

A second function performed by NAT, addressing extension, allows an organization to operate using RFC 1918 addresses behind a firewall and translate those addresses into a single "valid for public network usage" IP address. Because an organization might have hundreds or thousands of hosts behind a firewall, the reader may be curious as to how a firewall can accomplish the previously descried translation process. The answer to this is the use of high port addresses. For example, consider the network address 198.78.46.1 assigned to the public side of a firewall. Each host requiring access to the Internet would have its IP source datagram address converted into IP address 198.78.46.1 so that the firewall can differentiate datagrams flowing back to different IP addresses that were created by different hosts behind the firewall. The firewall will assign distinct source port numbers to the UDP or TCP header in each datagram and update a table of IP address/port-number assignments. Then, when a datagram is returned from the Internet to the firewall, it will search the IP address/port-number assignment table, retrieve the original source address, and form a new datagram so that it can flow to the correct destination. Thus, NAT provides both a security feature as well as address extension capability. Because IPv4 addresses are becoming quite scarce, many organizations view the ability of a firewall to perform NAT as an extremely important criterion, although its security function is not an important as the other firewall features previously described in this section.

Chapter 9

Emerging Technologies

The widespread adoption of the TCP/IP protocol suite by business, government, and academia has resulted in a significant amount of development effort being devoted to this protocol suite. This development effort recognizes that Internet usage has grown exponentially, from a few hundred thousand users at the beginning of the 1990s to over 100 million users at the beginning of the new millennium. With this vast market of Internet users, the acquisition of equipment by Internet service providers (ISPs) also became an extensive market for hardware and software developers. With the vast amount of funds that ISPs are expending in providing support to their customer base, they are also favorably viewing the development of new technologies that they could use for additional billing, enhancing customer retention, and differentiating their service from other ISPs. Thus, there is a receptive market from both end users and ISPs for the use of new technologies that add functionality to the TCP/IP protocol suite.

This chapter focuses on four emerging technologies related to the TCP/IP protocol suite: virtual private networking, mobile IP, Voice over IP, and IPv6. Although each of these technologies has been in existence for a number of years, they can be considered emerging technologies as their potential use is now being recognized, and products are rapidly being developed that utilize these technologies.

Virtual Private Networking

In the wonderful world of the TCP/IP protocol suite, virtual private networking and the creation of virtual private networks (VPNs) is being driven by the growth of the Internet and economics. With the rapid growth of the Internet, this global network now provides connectivity to just about every city on the globe. Thus, the use of the Internet represents a potential replacement for

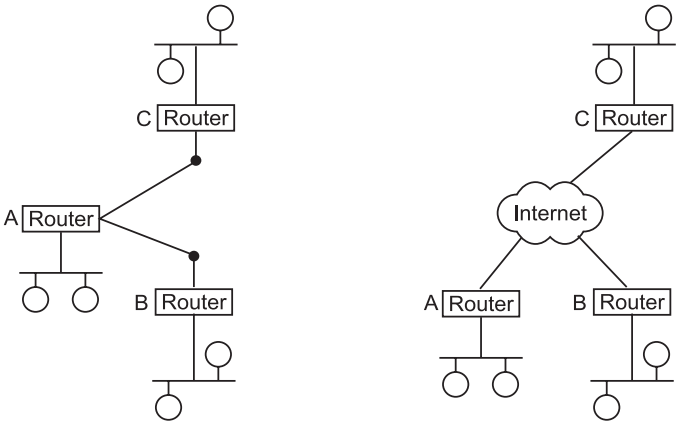


Exhibit 9.1. Comparing a Leased Line-Based Network versus the Use of a VPN over the Internet

expensive leased lines used by many organizations to interconnect geographically separated locations.

Benefits

To understand the benefits associated with the use of a VPN created via transmitting data over the Internet, one can compare and contrast the use of a private leased line-based network to the Internet. The left portion of Exhibit 9.1 illustrates a three-location, leased line-based network. The right portion of Exhibit 9.1 illustrates the use of a VPN created over the Internet to provide connectivity between the three locations.

Reducing Hardware Requirements

In examining Exhibit 9.1, note that the use of a private network requires more router ports than the creation of a VPN to interconnect three or more geographically separated locations. The use of a private network requires each location to be interconnected to other locations. To do so, the private network designer may have to interconnect some locations to multiple locations, such as connecting location A to locations B and C. In comparison, when a packet network such as the Internet is used, each location requires only one connection to the network and can use the routing capability of the packet network to access multiple locations via one serial connection. Thus, the use of the Internet can reduce hardware costs associated with obtaining connectivity, such as the number of router ports required for connecting three or more locations together.

Reliability

In addition to hardware savings, the use of a VPN via a packet network enables an organization to obtain the ability to use the mesh structure of the packet network. This means that if a router or transmission line within the Internet should fail, alternate routing within the Internet may alleviate the problem, providing additional reliability. Although a private organization can construct a mesh-structured network to obtain a similar degree of additional reliability, this is not common due to the high cost associated with establishing this type of network structure.

Economics

Although the ability to use less hardware and to obtain long-distance reliability are important advantages associated with a VPN, they are not the only advantages. Perhaps the key advantage is the ability of organizations to interconnect geographically separated locations under certain situations for a fraction of the cost of using leased lines. To illustrate the potential economic savings associated with the use of a VPN in comparison to a leased line-based network, reconsider Exhibit 9.1.

Assume that the private three-node network illustrated in the left portion of Exhibit 9.1 results in location A being 500 miles from location B and a similar distance from location C. Thus, the network would have a total of 1000 miles of leased lines. If one assumes that those leased lines are T1 circuits that operate at 1.544 Mbps and the monthly cost of each circuit is $4 per mile, then the long-distance cost of the leased line network becomes 1000 miles at $4 per mile per month, or $4000 per month.

In looking at the VPN created via the Internet shown in the right portion of Exhibit 9.1, one can see that the use of the Internet is distance insensitive for corporate users. The only charge is an access fee that connects each location via an Internet service provider (ISP) to the Internet. If it is furher assumed that each of the three locations is within a city or surrounding suburban area, then the monthly fee charged by an ISP to support T1 access can be expected to range between $1000 and $1500 per location. This means that to provide three locations with T1 Internet access, the monthly cost would range between $3000 and $4500. Thus, while it might be possible to save some money, it is also possible that the use of the Internet can result in an additional expenditure of funds versus a private leased line-based network. By now the reader might be a bit confused because this author previously mentioned that economics is the driving force for the creation of a VPN via the Internet. And because this author does not want readers to be confused, assume now that the three locations shown in the left portion of Exhibit 9.1 are New York City, Miami, and San Francisco.

Based on the previously mentioned revised locations, the distance between the three locations has now expanded to approximately 5000 miles. Thus, the

monthly cost of a private leased line-based network using T1 circuits would now become 5000 miles at $4 per mile per month, or $20,000 per month. Because the cost of accessing the Internet in major metropolitan areas is the same, the cost associated with connecting each location to the other via a VPN over the Internet remains the same. Thus, the cost would remain between $3000 and $4500 per month.

The preceding revision illustrates that the distance-insensitive cost associated with the use of the Internet can result in considerable savings when locations to be interconnected are relatively far apart. This also means that the further distant one location is from another, the greater the possible monthly cost savings. This also means that the global reach of the Internet could provide considerable economic savings for multinational organizations because they use international circuits that are relatively expensive in comparison to leased lines installed within a country. Given an appreciation for the benefits associated with the use of VPNs, one must also be aware of some of the limitations associated with the technology.

Limitations

Although not thought of as such each time a user transmits or receives e-mail via the Internet, that user employs a VPN that was created on a temporary basis. In fact, the use of VPNs by corporations follow a similar structure, with routing occurring on a packet-by-packet basis through the Internet. Unlike the transmission of conventional e-mail that might represent an update about family life or another personal matter, a VPN used by a company requires several features that are not normally needed on a personal messaging basis. Those features include authentication and encryption, as well as the use of a firewall because corporate information will now be flowing through the Internet. This opens the corporate network to potential attack from the Internet community of users.

Authentication

Authentication represents the process of verifying that a person is the person he or she claims to be. As discussed in Chapter 8, one of the most popular methods of authentication is based on the use of token generating cards. Regardless of the type of authentication used, one must factor its cost into consideration to determine the true cost of using a VPN versus a leased line network. Normally, authentication is not required on a leased line network as the network connects fixed locations. Thus, a leased line network commonly represents a “closed” network that outsiders cannot access. In comparison, the Internet represents an “open” network. Once a person notes the host or IP address of a device, that person can attempt to transmit information to that device.

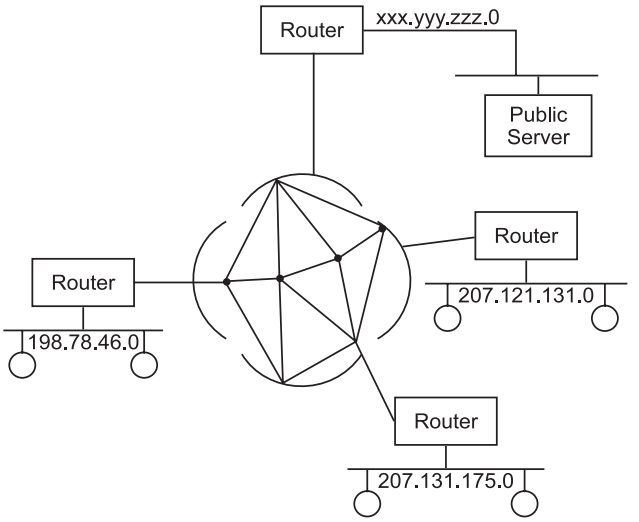


Exhibit 9.2. When Encryption is Used on a VPN, the Destination of Packets Must be Examined as a Basis for Determining Whether or Not to Encrypt the Packet

Encryption

Because it is doubtful that an organization would allow vital corporate information to flow clear across the Internet where it could possibly be intercepted, another commonly required VPN is encryption. However, VPN encryption is both more difficult to perform and more expensive than conventional encryption products used to secure transmission via leased lines.

The reason VPN encryption is more complex and costly than encryption performed on point-to-point leased lines results from the fact that routing via a VPN is more complex. For example, consider Exhibit 9.2, which illustrates the assignment of Class C IP addresses to each of three networks to be interconnected via the Internet. Note that a fourth network on which a public server resides, such as www.whitehouse.com, is shown with the network address xxx.yyy.zzz.0 to indicate that it can be any address other than the three network addresses of the geographically separated networks to be interconnected via a VPN. Because a user on network 198.78.46.0 may periodically surf the Web while at other times access a server on network 207.121.131.0 or network 205.131.175.0, the device performing encryption must be configured to distinguish a VPN destination address. In addition, because one would not want to use the same encryption key between all sites, the equipment must support multiple keys.

Other Issues to Consider

In addition to security, there are two additional issues one must consider when thinking about the use of a VPN. Those issues are management control

and the latency or delay through the Internet. Concerning management control, unlike the use of leased lines that can contact a single communications carrier in an attempt to resolve a problem, if a problem occurs on the Internet, one's ability to communicate with a carrier is restricted when dealing with an ISP. Concerning latency, because other traffic carried on the Internet is not predictable, the delay packets experience will be random. This means that certain delay-sensitive applications, such as real-time command and control for numeric machinery as well as Voice over IP, may or may not be suitable for a VPN. Despite such problems, VPNs represent an emerging technology, with support even included in Microsoft's popular Windows NT server. Thus, this section concludes with a discussion of how one can set up an NT server to allow dial-up access that in turn permits a remote user to connect to the organization's internal network. If that network is in turn connected to the Internet, one can then provide employees with the ability to access other organizational locations via a local telephone call even if those locations are thousands of miles away or are on another continent.

Setting Up Remote Access Service

Exhibit 9.3 illustrates how one would begin to install Microsoft's Remote Access Service on a Windows NT server. In this example, one would enter Start > Settings > Control Panel and then select Network. Once the Network dialog box is displayed, click on the Services tab in the box and the Add button located on that tab.

By performing the previously described operations, a dialog box labeled "Select Network Service" will be displayed. Exhibit 9.4 illustrates an example of that dialog box. Note that the Remote Access Service entry is shown highlighted. Click on the button labeled "OK" and Windows will prompt you for a disk or a CD that is provided with NT. After entering Windows NT, CD RAS will be both located and a portion of the program will be installed.

After RAS is installed, different devices can be added that will allow access to services. This is shown in Exhibit 9.5, where the author has added a generic 28800 bps modem to support dial-in service. Exhibit 9.6 shows the result of the previous action, with the modem installed on the COM1 serial communications port.

By examining the right portion of Exhibit 9.6, one notes a button labeled "Network." Clicking on this button provides the ability to allow a remote client to use NetBEUI, TCP/IP, and IPX protocols. This capability is shown in Exhibit 9.7. Note that this dialog box also provides the capability to require authentication and encryption for the remote user.

If one clicks on the Configure box next to the TCP/IP option, one obtains the ability to configure a variety of TCP/IP options. Exhibit 9.8 illustrates the resulting dialog box, labeled "RAS Server TCP/IP Configuration." Note from Exhibit 9.8 that one can enable RAS service for the entire network or the server. Also note that one can provide a dynamic or static IP address to remote clients or even allow them to request a predefined IP address. Thus, Microsoft's

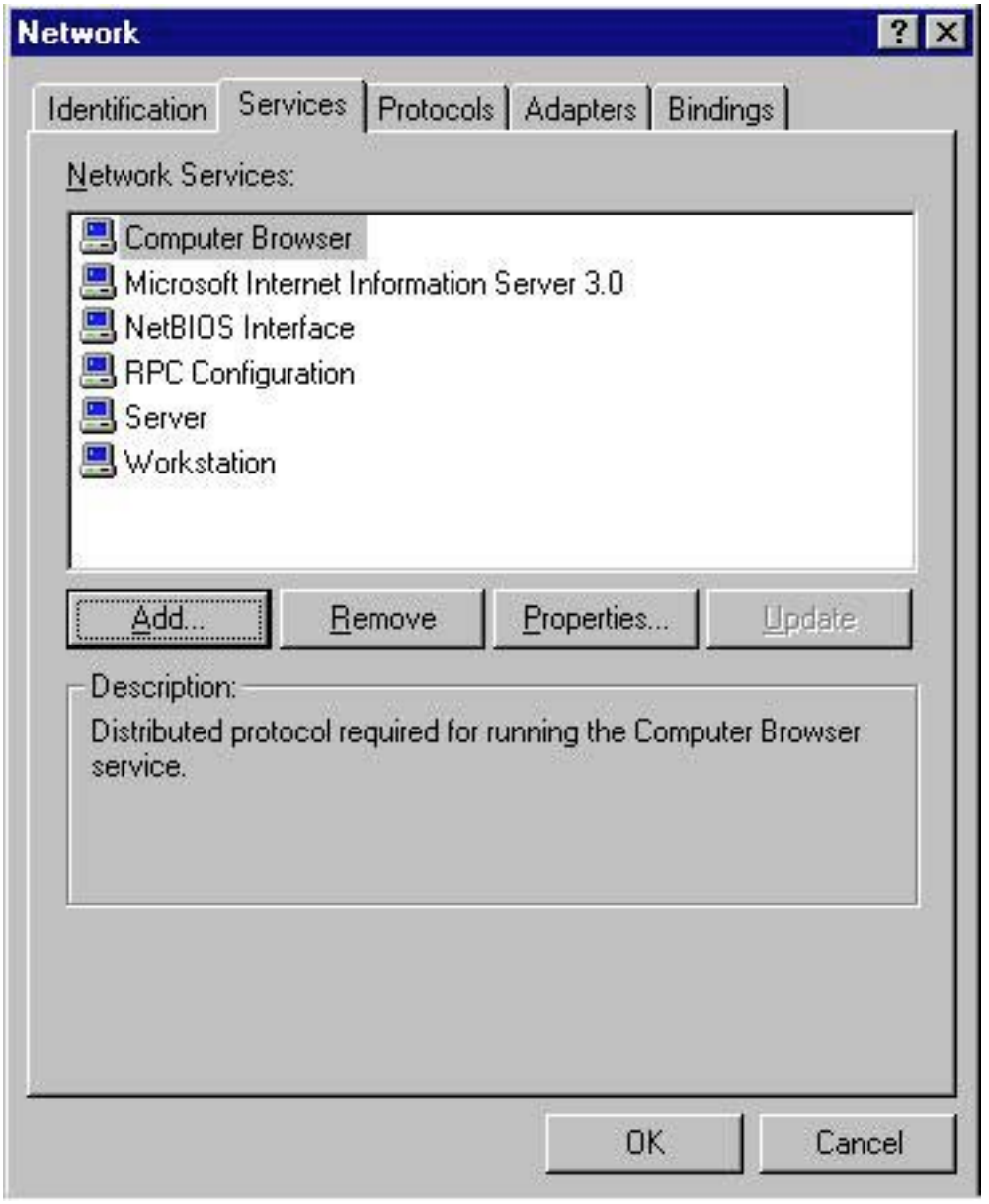


Exhibit 9.3. Installing Remote Access Services on an NT Server Requiring Selection of Services Tab from the Network Dialog Box

Remote Access Service server provides support for different addressing schemes. When combined with other Microsoft products, Windows NT servers can be used for routing, which allows an organization to spend a few thousand dollars per location to interconnect geographically separated networks via the Internet. Thus, many products are appearing in the marketplace that provide a VPN capability and represent another driving force for adopting VPNs.

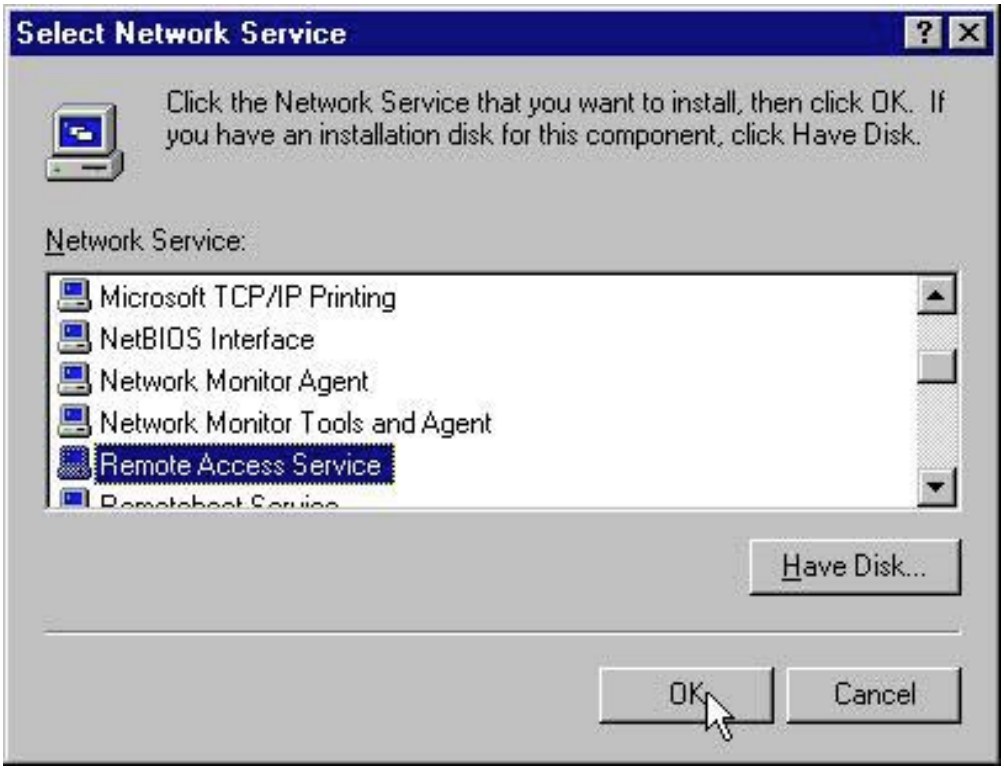

Select Network Service
Click the Network Service that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.
Network Service:
Microsoft TCP/IP Printing
NetBIOS Interface
Network Monitor Agent
Network Monitor Tools and Agent
Remote Access Service
Have Disk...
OK
Cancel

Exhibit 9.4. Installing RAS from the Select Network Service Dialog Box

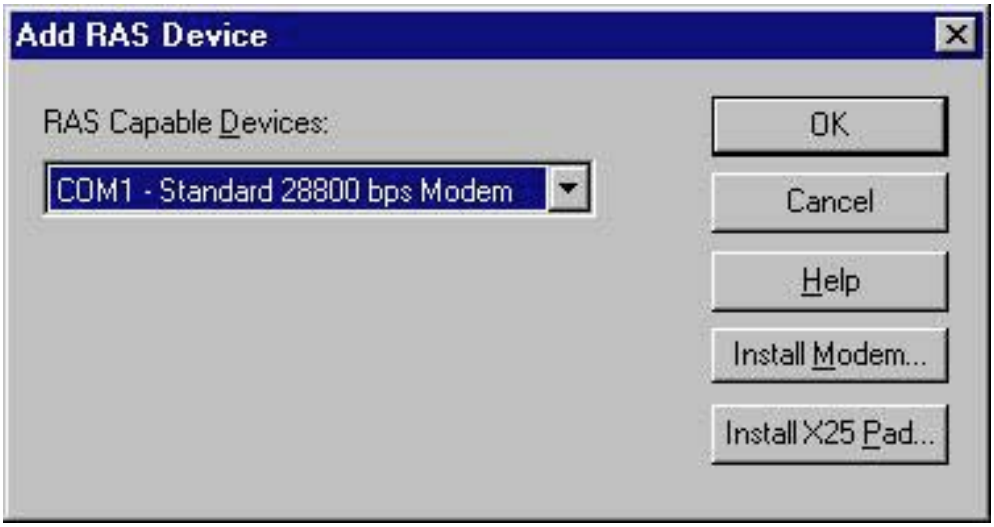


Exhibit 9.5. Adding a Generic 28,800-bps Modem to Support Dial-Up Access to a RAS Server

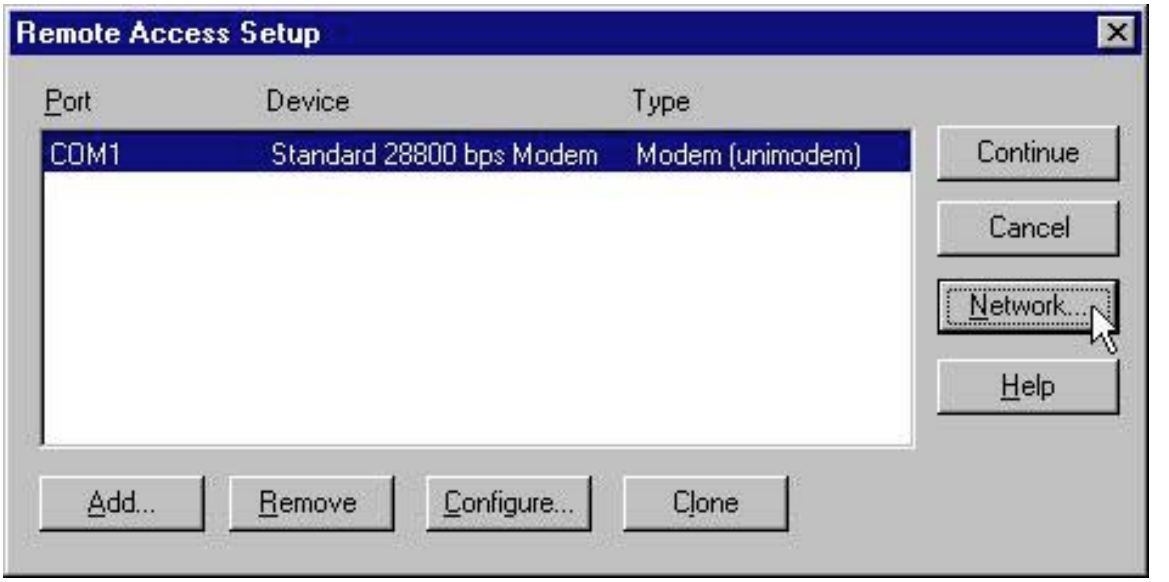


Exhibit 9.6. Examining the Remote Access Setup Dialog Box after Dial-In Access Is Configured

Mobile IP

Although a mobile person can access their home location via the dial-up services of an ISP, there is no method for host computers on an internal corporate network to easily note the mobile user's identity and imitate communications with the person's computer. Because of this, a TCP/IP network can be considered to represent a client-driven network, with the client having to initiate communications sessions.

With the growth in the use of cell phones and the development of a limited e-mail and browsing capability by those devices, it becomes possible to employ server-side session initiation. That is, a server can be configured to interface the telephone network and transmit messages to mobile cell phone subscribers. However, a similar mechanism is not available for the traveling notebook operator who may be required to check several mail systems to determine what messages, if any, are awaiting his or her action. Recognizing this problem was probably a contributing factor to the development of mobile IP, which represents a second emerging technology to be discussed in this chapter.

Overview

Under mobile IP, a user makes a connection to his home network and registers his presence with a mobile IP server. If the remote user is using the services of an ISP and has a temporary IP address, that address is also registered with

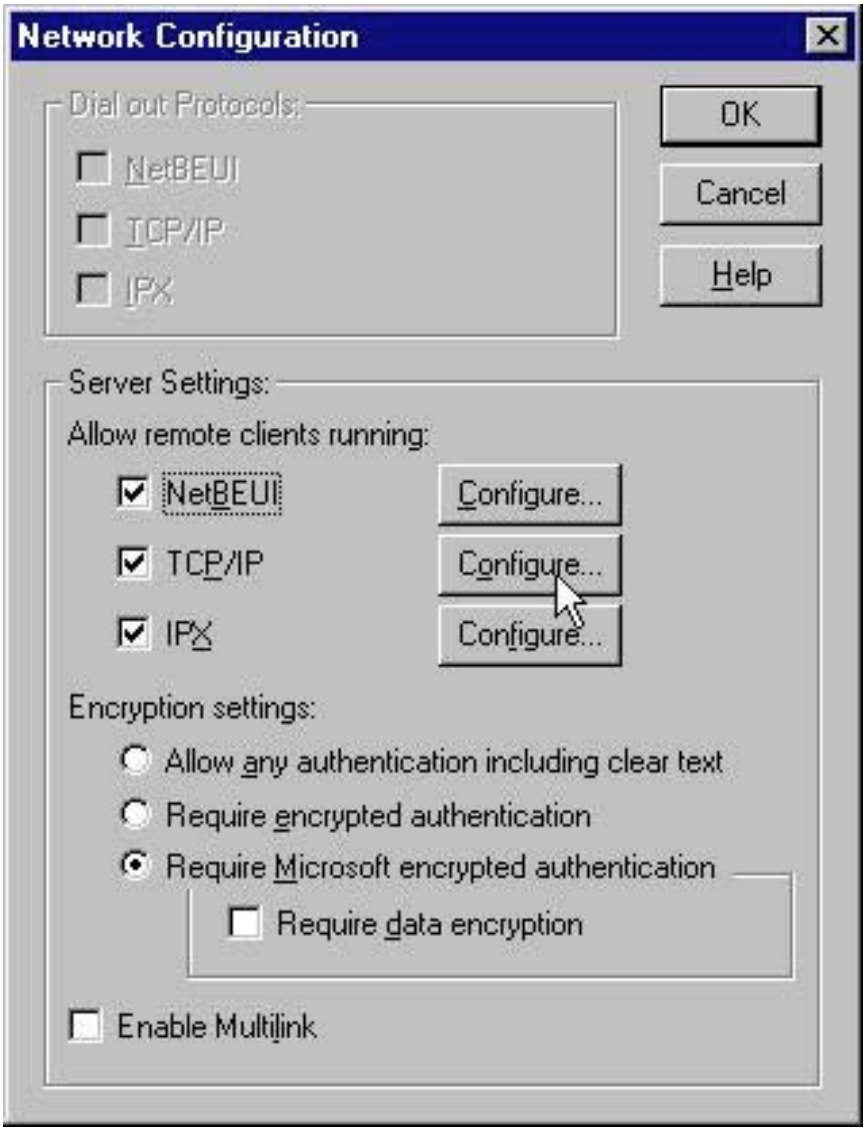


Exhibit 9.7. Specifying the Protocols the Server Will Support and the Use of Authentication and Encryption through the Network Configuration Dialog Box

the mobile IP server. A second feature or function of that server is to serve as a focal point for applications that need to communicate with the mobile user. Thus, it becomes possible for e-mail and other server-side applications to note the presence of a mobile IP user and communicate with the user through the services of the mobile IP server.

Operation

Exhibit 9.9 provides an example of how a mobile IP server might be used. In this example, assume that the mobile user was previously registered with

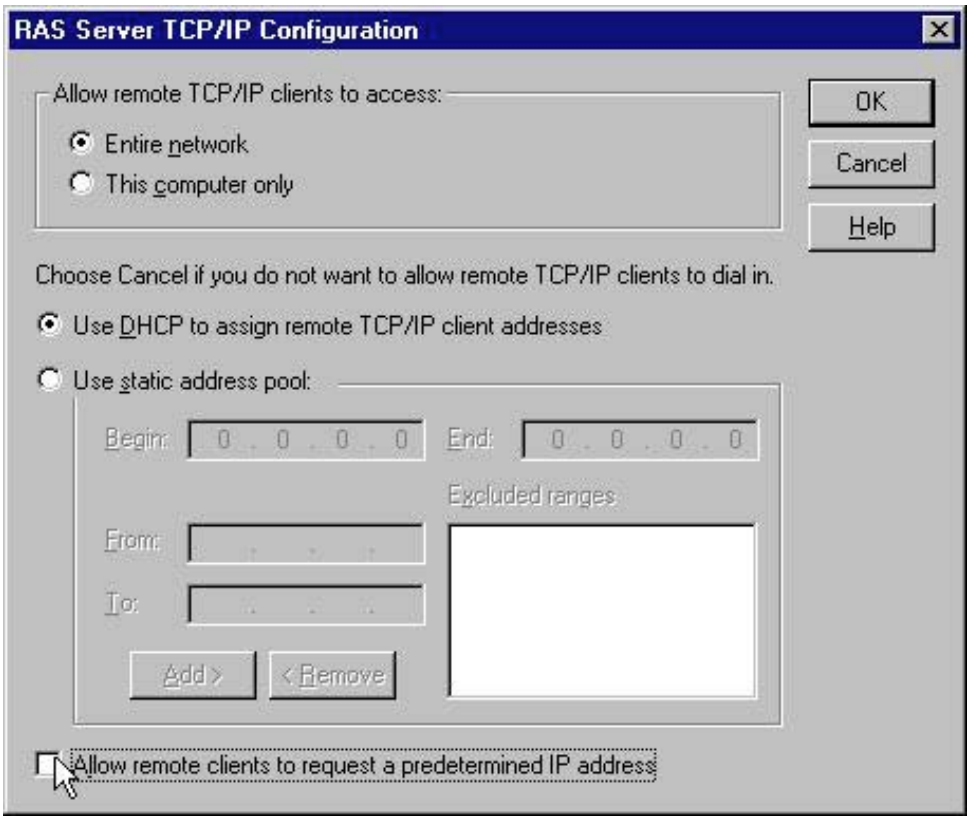


Exhibit 9.8. RAS Server TCP/IP Configuration Dialog Box

the server, and applications that need to reach the user to know his or her presence will be established via notification from the server.

On a trip to Japan, the traveling executive dials the Internet via the hotel where he or she is staying. While online, the ISP serving Japan assigns a temporary IP address to the user. This address is noted by the mobile IP server (1), which informs predefined applications that the distant user is online. This is illustrated by (2). Next, each application, such as e-mail (3) and digitized voice mail (3), uses the temporary IP address to establish communications with the traveling executive.

While the concept behind mobile IP has been around for a few years, until recently it was easier for a person to simply communicate with his or her e-mail system than support server-side initiation. Because there are emerging applications beyond e-mail that people may wish to access, it may be easier to allow the applications to contact the user when they access the home network than to require persons to check numerous servers.

Voice over IP

The transmission of voice over an IP network, which is referred to as Voice over IP, represents an evolving technology that offers both individuals and

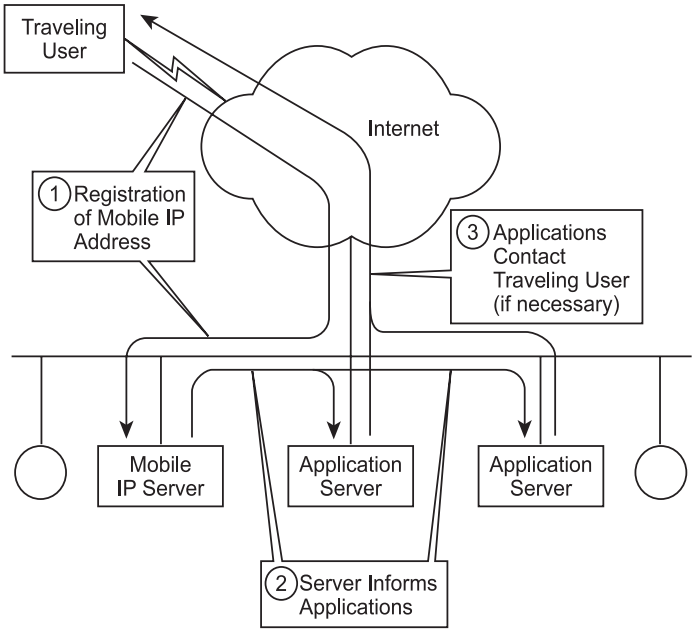


Exhibit 9.9. Mobile IP in Operation

organizations the potential to realize substandard economic savings. In fact, if one can obtain a voice over IP transmission capability by upgrading existing equipment through the installation of voice modules on a router or the use of a getaway on an existing network infrastructure, it becomes possible to transmit a digitized voice call over an intranet or the Internet for as little as 0.1 cent per minute. This cost is low enough to make the Sprint dime lady blush and explains the key interest of organizations in applying this technology.

Constraints

There are several key constraints governing the ability of digitized voice to be transmitted over an IP network and successfully reconstructed at the destination. Those constraints include end-to-end latency, the random nature of packet networks, the voice digitization method used, and the need to subdivide packets containing data into minimum-length entities. Each of these constraints is interrelated.

Latency

Latency or end-to-end delay governs the ability of reconstructed voice to sound normal or awkward. The total one-way delay that a packet experiences as it flows from source to destination via an IP network depends on several factors. Those factors include the speed of the ingress and egress lines from locations

connected to the Internet or a private intranet, the voice coding algorithm used to digitize voice, the number of router hops through the Internet or a private intranet from source to destination, and the activity occurring at and between each hop. When all of these factors are considered as an entity, the one-way delay of a packet should not exceed 250 ms and should probably be under 200 ms to obtain a good quality of reconstructed voice.

Because delays on the Internet can easily exceed 200 ms without considering the voice coding algorithm delay, this explains why it is not possible to consider the Internet as fully ready to support digitized voice at this time. Because it is relatively easy but costly to add bandwidth to a private intranet, one can do so to reduce delays. However, the additional cost associated with reducing latency can increase the cost of transporting Voice over IP. In the near future, the hundreds of thousands of miles of fiber optic cable being installed throughout the United States, western Europe, and other locations around the globe should result in an increase in transmission capacity by several orders of magnitude over existing transmission facilities. As ISPs upgrade their backbones, it may be possible within a few years for the problem of latency to be considerably reduced in comparison to the role it plays today in hampering Internet telephony from achieving widespread acceptance.

Packet Network Operation

The operation of a packet network can be considered to represent a random process, with data arriving at routers occurring on a random basis. This means that the delay experienced by a series of packets transporting digitized voice will not only result in a random transit delay through the network, but will also in addition result in random delays between packets. This also means that the ability to transport digitized voice and reconstruct it so that it sounds natural requires the use of a "jitter buffer."

A jitter buffer represents a small portion of memory at the recipient equipment that temporarily stores received packets transporting voice. To ensure that packets are extracted in their appropriate time sequence, each packet must be timestamped. In a Voice over IP environment, most applications currently use a protocol referred to as the Real Time Protocol (RTP). RTP provides the capability to both timestamp and sequence number packets. RTP is commonly implemented over UDP. UDP is used to transmit digitized voice because no error checking is necessary, as erroneous packets cannot be retransmitted. Although RTP is important for the appropriate use of a jitter buffer, this new header adds a bit of delay to the flow of packets. In addition, the use of a jitter buffer for temporary data storage also adds to end-to-end delay. Although many jitter buffers are capable of being configured to store from 0 (not operational) to 255 ms of speech, the wider the buffer, the greater the potential delays to voice transporting packets. Thus, the configuration of a jitter buffer must be considered with respect to other potential delays as well as the voice digitization coding method to be used.

Voice Digitization Method

There are currently over half a dozen voice digitization methods supported by most Voice over IP hardware products. These methods vary with respect to the coding rate they operate at, the coding delay, and the legibility of reconstructed voice. The latter is normally specified by a subjective measurement referred to as Mean Optimum Score (MOS). In general, the lower the data rate, the higher the coding delay and the lower the MOS score. For example, Pulse Code Modulation (PCM), which is used extensively on the Public Switched Telephone Network (PSTN), operates at 64 Kbps, has a coding delay of approximately 1 ms, and has the highest MOS of all coding methods. In comparison, a popular voice coder used on packet networks referred to as G.723 operates at 5.3 Kbps or 6.2 Kbps, but has a coding delay of 30 ms and a much lower MOS score than PCMs.

Exhibit 9.10 lists four popular voice coding methods. The first two, Pulse Code Modulation (PCM) and Adaptive Differential Pulse Code Modulation (ADPCM), represent waveform coding techniques primarily used on the PSTN (PCM) and on international circuits (ADPCM) over which public network calls are routed between countries. Both coding methods are referred to as "toll quality" and represent the sound of reconstructed voice for which other methods are commonly compared. The algorithm delay for coding a voice sample via PCM or ADPCM is very fast, typically 1 ms. However, the resulting bit rate is relatively high in comparison to recently developed voice coding methods that use both waveform sampling and speech synthesis — a technique that is referred to as hybrid coding.

Exhibit 9.10 Common Speech Coding Algorithms

Standard	*Coding Method*	*Bit Rate*	*Delay*	*MOS*
G.711	PCM	64 Kbps	1 Φs	4.4
G.726	ADPCM	32 Kbps	1 Φs	4.4
G.728	LD-CELP	16 Kbps	10 ms	4.2
G.723.1	MP-MLQ	5.3/6.3 Kbps	30 ms	3.9

Two popular hybrid voice coding techniques standardized by the ITU are G.728 and G.723.1, the latter a multi-rate coder. G.728 is a low-delay coder, with the algorithm requiring 10 ms. This delay is still a thousand times greater than the delay associated with PCM and ADCPM. Also note that the G.723.1 standard is 30 ms, which can represent a considerable period of time when overall end-to-end delay to include the coding algorithm is limited.

Because end-to-end delay must be less than 250 ms and preferably below 200 ms, many times the use of a very low bit rate voice digitization technique will result in an excessive amount of delay. If one's equipment supports multiple coders, one technique to consider to enhance the quality of reconstructed voice is the use of a different coder.

A second problem concerning the use of voice coders is the fact that at the present time there are not any standards that enable equipment produced by different vendors to negotiate the use of a voice coder. Although the Frame Relay Forum developed a standard for Voice over Frame Relay during 1997, a similar standard is still missing for use on TCP/IP networks. Thus, equipment interoperability can be considered in its infancy, and organizations may have to experiment using different coders to select an optimum one based upon a series of factors to include routing delays through the packet network.

Packet Subdivision

As indicated earlier in this book, a datagram can be up to 65,535 bytes in length. Unfortunately, if a long datagram should flow between two datagrams transporting digitized voice, the lengthy datagram can introduce a significant delay that makes the reconstruction of quality-sounding voice difficult, if not impossible. Due to this, it is necessary to use equipment to limit the length of packets transporting data. Unfortunately, there are two constraints associated with packet subdivision. First, one can only use equipment at the entrance to a packet network to subdivide lengthy packets. This means that one cannot control lengthy packets transmitted by other organizations through the packet network. Second, by taking one lengthy packet transporting data and subdividing it into two or more packets, one increases network traffic and reduces transmission efficiency. This results from the fact that one now has more overhead in the form of multiple headers rather than one header. This additional traffic can overtax routers and result in routers doing what they are programmed to do under an overload condition — drop packets. Thus, it is entirely possible that the medicine in the form of packet subdivision could kill the patient. Despite the previously mentioned problems, Voice over IP is a viable emerging technology. To understand why organizations are excited concerning the use of this technology, one can explore two networking configurations suitable for consideration on the Internet or an intranet, assuming end-to-end latency can be obtained at an acceptable level.

Networking Configurations

Two of the most popular methods for transporting voice over an IP network are through the use of voice modules installed in a router and the use of a stand-alone voice gateway. This section examines the use of each method.

Router Voice Module Utilization

Recognizing the fact that many organizations can benefit from the convergence of voice and data and that they already operate private IP networks, several router manufacturers have introduced voice modules that are designed for

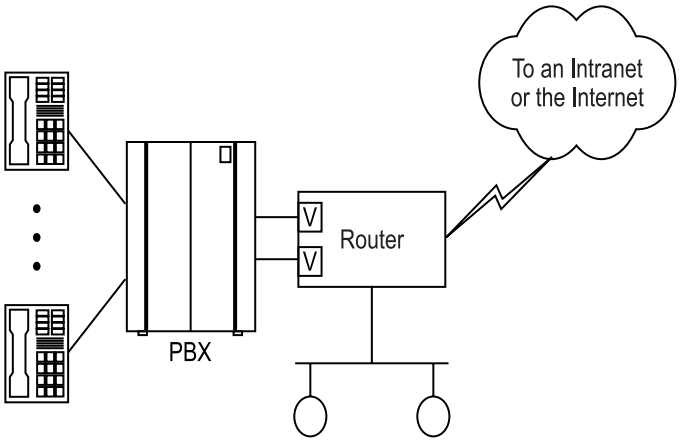


Exhibit 9.11. Using Router Voice Modules to Obtain an Integrated Voice/Data Networking Capability

installation within different router products. While voice modules can vary in capability and functionality between vendors, they perform a set of common features. These features include providing an interface to different signaling methods associated with the direct connection of PBX ports or individual telephone instruments, supporting several voice coding algorithms, and prioritizing the flow of datagrams transporting voice into router queues for faster placement onto a serial communications line. Exhibit 9.11 illustrates the potential use of voice digitization modules installed in a router to provide the ability to transmit both voice and data over a common network infrastructure.

In examining Exhibit 9.11, note that an organization would program the PBX to establish a new prefix for voice users to access the router's voice modules. For example, dialing a "6" might route calls to the router, while dialing a prefix of "9" would be used to connect to an outside line for calling via the PSTN. One would also configure the voice modules via the router's operating system, such as selecting a specific voice coding algorithm and setting a specific priority for voice in comparison to data so that datagrams transporting voice will receive preference for extraction from the queue in the router for transmission onto the serial port connecting the router to a private intranet or the public Internet.

Voice Gateway

A second common method to consider for integrating the transmission of voice and data over an IP network is obtained through the use of a voice gateway. Exhibit 9.12 illustrates the potential use of this communications device.

In examining Exhibit 9.12, one would interface a voice gateway to a PBX similar to the manner by which one would connect a PBX to voice modules installed in a router. Unlike the use of a router where packets transporting

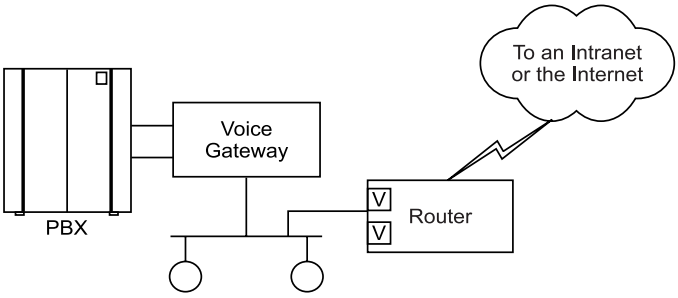


Exhibit 9.12. Using a Voice Gateway

digitized voice do not flow on a LAN, when a gateway is used, the packets flow on the local network. Thus, take care to ensure that the utilization level of the network is not over 50 percent. Otherwise, an excessive amount of collisions could occur that add to the latency of packets transporting voice.

A key advantage to the use of stand-alone voice gateways over the use of voice modules installed in routers is the fact that the former scale better than the latter. For example, current voice gateways are obtainable that typically support 4, 8, 16, 32, 64, and 128 voice ports, with expansion possible on some getaways that allow up to 1024 ports to be supported. In comparison, many routers only support the addition of two to four voice modules, with each module capable of supporting a limited number of ports.

Although a few routers now offer voice modules that can be directly interfaced to a digital ISDN port on a router and can directly accept 24 voice calls, such modules are only supported on high-end routers whose basic cost can exceed $50,000. In comparison, low-end routers that support a limited number of voice connections may have pricing that begins at $2500 and represents a more viable solution for integrating voice and data at a branch office. Low-end routers have a limited scaling capability, and one may wish to carefully consider the use of a voice gateway that has an expansion capability. If an organization uses LAN switches rather than a shared media network, it then becomes possible to connect the gateway to a switch port and avoid potential latency problems associated with the use of shared media networks.

IPv6

This section on emerging technologies, concludes with a discussion of the new version of the Internet Protocol, IPv6. As noted earlier in this book, the near-exponential growth in the use of the Internet has rapidly depleted the quantity of available IP network addresses and has resulted in such addresses becoming a precious commodity. In examining IPv6, note that its addressing capability ensures that every man, woman, and child on the planet — as well as every electronic device — can obtain an IPv6 address. This capability

provides a mechanism to enable the development of intelligent network-based home appliances and other devices that could be managed by a service organization or the homeowner from their office. Thus, IPv6 can be considered to provide a foundation for extending the capabilities of the Internet to new applications that can be expected to arise during the new millennium.

Overview

IPv6 was developed as a mechanism to simplify the operation of the Internet Protocol, provide a mechanism for adding new operations as they are developed through a header daisy-chain capability, add built-in security and authentication, and extend source and destination addresses to an address space that could conceivably meet every possible addressing requirement for generations. The latter is accomplished through an expansion of source and destination addresses to 128 bits and is the focus of this section.

Address Architecture

IPv6 is based on the same architecture used in IPv4, resulting in each network interface requiring a distinct IP address. The key differences between IPv6 and IPv4 with respect to addresses are the manner by which an interface can be identified, and the size and composition of the address. Under IPv6, an interface can be identified by several addresses to facilitate routing and management. In comparison, under IPv4, an interface can only be assigned one address. Concerning address size, IPv6 uses 128 bits, or 96 more bits than an IPv4 address.

Address Types

IPv6 addresses include unicast and multicast, which were included in IPv4. In addition, IPv6 adds a new address category known as anycast. Although an anycast address identifies a group of stations similar to a multicast address, a packet with an anycast address is delivered to only one station, the nearest member of the group. The use of anycast addressing can be expected to facilitate network restructuring while minimizing the amount of configuration changes required to support a new network structure. This is because one can use an anycast address to reference a group of routers, and the alteration of a network when stations use anycast addressing would enable them to continue to access the nearest router without a user having to change the address configuration of their workstation.

Address Notation

Because IPv6 addresses consist of 128 bits, a mechanism is required to facilitate their entry as configuration data. The mechanism used is to replace those bits

with eight 16-bit integers separated by colons, with each integer represented by four hexadecimal digits. For example:

6ACD:00001:00FC:B10C:0001:0000:0000:001A

To facilitate the entry of IPV6 addresses, one can skip leading zeros in each hexadecimal component. That is, one can write 1 instead of 0001 and 0 instead of 0000. Thus, this ability to suppress zeroes in each hexadecimal component would reduce the previous network address to:

6ACD:1:FC:B10C:1:0:0:1A

Under IPv6, a second method of address simplification was introduced, the double-colon (::) convention. Inside an address, a set of consecutive null 16-bit numbers can be replaced by two colons (::) Thus, the previously reduced IP address could be further reduced to:

6ACD:1:FC:B10C:1::1A

It is important to note that the double-colon convention can only be used once inside an address. This is because the reconstruction of the address requires the number of integer fields in the address to be subtracted from eight to determine the number of consecutive fields of zero value the double-colon represents. Otherwise, the use of two or more double-colons would create ambiguity that would not allow the address to be correctly reconstructed.

Address Allocation

The use of a 128-bit address space provides a high degree of address assignment flexibility beyond that available under IPv4. IPv6 addressing enables Internet service providers to be identified as well as includes the ability to identify local and global multicast addresses, private site addresses for use within an organization, hierarchical geographical global unicast addresses, and other types of addresses. Exhibit 9.13 lists the initial allocation of address space under IPv6.

The Internet Assigned Numbers Authority (IANA) was assigned the task of distributing portions of IPv6 address space to regional registries around the world, such as the InterNIC in North America, RIPE in Europe, and APNIC in Asia. To illustrate the planned use of IPv6 addresses, the discussion continues with what will probably be the most common type of IPv6 address — the provider-based address.

Provider-Based Addresses

The first official distribution of IPv6 addresses will be accomplished through the use of provider-based addresses. Based on the initial allocation of IPv6

Exhibit 9.13 IPv6 Address Space Allocation

Allocation	*Prefix (binary)*	*Fraction of Address Space*
Reserved	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP allocation	0000 001	1/128
Reserved for IPX allocation	0000 010	1/128
Unassigned	0000 011	1/128
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Unassigned	001	1/8
Provider-based Unicast Address	010	1/8
Unassigned	011	1/8
Reserved for Geographic-based Unicast Address	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-Local Use Addresses	1111 1110 10	1/1024
Site-Local Use Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

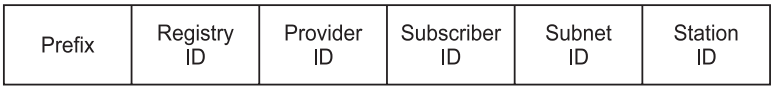
addresses as shown in Exhibit 9.13, each provider-based address will have the three-bit prefix 010. That prefix will be followed by fields that identify the registry that allocated the address, the service provider, and the subscriber. The latter field actually consists of three sub-fields: a subscriber ID that can represent an organization, and variable network and interface identification fields used in a similar manner to IPv4 network and host fields. Exhibit 9.14 illustrates the initial structure for a provider-based address.

Special Addresses

Under IPv6, five special types of unicast addresses were defined, of which one deserves special attention. That address is the Version 4 address, which was developed to provide a migration capability from IPv4 to IPv6.

In a mixed IPv4/IPv6 environment, devices that do not support IPv6 will be mapped to version 6 addresses using the following form:

0:0:0:0:0:FFFF:w.x.y.z



Legend:

Prefix	three bits set to 010
Registry	5 bits identify organization that allocated the address
Provider	24 bits with 16 used to identify ISP and 8 used for future extensions
Subscriber	32 bits with 24 used to identify the subscriber and 8 for extension
Subnet	16 bits to identify the subnetwork
Station	48 bits to identify the station

Exhibit 9.14. Provider-Based Address Structure

Here, w.x.y.z represents the original IPv4 address. Thus, IPv4 addresses will be transported as IPv6 addresses through the use of the IPv6 version 4 address format. This means that an organization with a large number of workstations and servers connected to the Internet only has to upgrade its router to support IPv6 addressing when IPv6 is deployed. Then, the network can be gradually upgraded on a device-by-device basis to obtain an orderly migration to IPv6.

Although IPv6 is being used on an experimental portion of the Internet, its anticipated movement into a production environment was delayed due to the more efficient use of existing IPv4 addresses. This occurred via network address translation, which was described in Chapter 8. While the use of IPv6 is less pressing than thought just a few years ago, no matter how efficient the allocation of the remaining IPv4 addresses becomes, it is a known fact that within the next few years, all available addresses will be used. Prior to that time, one can expect a migration to IPv6 to occur.

APPENDIXES: TCP/IP PROTOCOL REFERENCE NUMBERS

The appendixes provide a comprehensive reference to several key TCP/IP protocol numbers. As indicated earlier in this book, TCP/IP is not a single protocol. Instead, it represents a layered protocol that has several components.

One of the major components of TCP/IP includes the Internet Control Message Protocol (ICMP) used to convey different types of control messages within an IP datagram. Because of the flexible design of the ICMP format, different types of messages can be conveyed by altering the value of its TYPE field. In addition, many ICMP TYPE field values have a CODE field whose value further clarifies the type of message being conveyed. Thus, Appendix A is included in this book to provide a reference to ICMP TYPE and CODE values.

A second major component of the TCP/IP protocol suite is the Internet Protocol (IP). Because an IP datagram can transport an ICMP message, a TCP segment, or a UDP datagram, as well as other higher layer protocols, a mechanism is required to define the upper layer protocol being conveyed. That mechanism is provided by the Protocol Type field, which identifies the header following the IP datagram. Appendix B contains a listing of the values of the Protocol Type field.

A third major characteristic of the TCP/IP protocol suite is the use of port numbers to identify applications, enabling the multiplexing of different types of application-based traffic from and to common addresses. The use of port numbers explains how, for example, a Web server could also support FTP and telnet operations. The first 1024 port numbers, referred to as Well Known Ports, are summarized in Appendix C.

Appendix A

ICMP Type and Code Values

ICMP TYPE NUMBERS

The Internet Control Message Protocol (ICMP) has many messages that are identified by a "type" field are listed below.

Type	*Name*
0	Echo Reply
1	Unassigned
2	Unassigned
3	Destination Unreachable
4	Source Quench
5	Redirect
6	Alternate Host Address
7	Unassigned
8	Echo
9	Router Advertisement
10	Router Selection
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply
19	Reserved (for Security)
20–29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error

(continues)

Type	*Name (continued)*
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
37	Domain Name Request
38	Domain Name Reply
39	SKIP
40	Photuris
41–255	Reserved

Many of these ICMP types have a "Code" field. One can view the Code field as providing a specific clarifier for the value in the Type field. For example, a router generating a Type field value of 3 indicates the IP destination address was unreachable, but does not specifically indicate why it was unreachable. Here the Code field value would clarify the reason why the destination address was unreachable. The following table lists the Type field values and their applicable Code field values, if any:

Message Type	*Code Field Value*
0	Echo Reply **Codes** 0 No Code
1	Unassigned
2	Unassigned
3	Destination Unreachable **Codes** 0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated 9 Communication with Destination Network is Administratively Prohibited 10 Communication with Destination Host is Administratively Prohibited

Message Type	Code Field Value (continued)
	11 Destination Network Unreachable for Type of Service
	12 Destination Host Unreachable for Type of Service
	13 Communication Administratively Prohibited
	14 Host Precedence Violation
	15 Precedence cutoff in effect
4	Source Quench
	Codes
	0 No Code
5	Redirect
	Codes
	0 Redirect Datagram for the Network (or subnet)
	1 Redirect Datagram for the Host
	2 Redirect Datagram for the Type of Service and Network
	3 Redirect Datagram for the Type of Service and Host
6	Alternate Host Address
	Codes
	0 Alternate Address for Host
7	Unassigned
8	Echo
	Codes
	0 No Code
9	Router Advertisement
	Codes
	0 No Code
10	Router Selection
	Codes
	0 No Code
11	Time Exceeded
	Codes
	0 Time to Live exceeded in Transit
	1 Fragment Reassembly Time Exceeded
12	Parameter Problem
	Codes
	0 Pointer indicates the error
	1 Missing a Required Option
	2 Bad Length

(continues)

Message Type	*Code Field Value (continued)*
13	Timestamp **Codes** 0 No Code
14	Timestamp Reply **Codes** 0 No Code
15	Information Request **Codes** 0 No Code
16	Information Reply **Codes** 0 No Code
17	Address Mask Request **Codes** 0 No Code
18	Address Mask Reply **Codes** 0 No Code
19	Reserved (for Security)
20–29	Reserved (for Robustness Experiment)
30	Traceroute
31	Datagram Conversion Error
32	Mobile Host Redirect
33	IPv6 Where-Are-You
34	IPv6 I-Am-Here
35	Mobile Registration Request
36	Mobile Registration Reply
39	SKIP
40	Photuris **Codes** 0 Reserved 1 unknown security parameters index 2 valid security parameters, but authentication failed 3 valid security parameters, but decryption failed

Appendix B

Internet Protocol (IP) Protocol Type Field Values

In the Internet Protocol Version 4 (IPv4), eight 8-bit "Protocol" field is used to identify the next level protocol. The values of the Assigned Internet Protocol Numbers and associated protocols are listed in the following table.

Decimal	*Keyword*	*Protocol*
0	HOPOPT	IPv6 Hop-by-Hop Option
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
3	GGP	Gateway-to-Gateway
4	IP	IP in IP (encapsulation)
5	ST	Stream
6	TCP	Transmission Control
7	CBT	CBT
8	EGP	Exterior Gateway Protocol
9	IGP	Any private interior gateway (used by Cisco for their IGRP)
10	BBN-RCC-MON	BBN RCC Monitoring
11	NVP-II	Network Voice Protocol
12	PUP	PUP
13	ARGUS	ARGUS
14	EMCON	EMCON
15	XNET	Cross Net Debugger
16	CHAOS	Chaos
17	UDP	User Datagram
18	MUX	Multiplexing
19	DCN-MEAS	DCN Measurement Subsystems

(continues)

Decimal	Keyword	Protocol (continued)
20	HMP	Host Monitoring
21	PRM	Packet Radio Measurement
22	XNS-IDP	XEROX NS IDP
23	TRUNK-1	Trunk-1
24	TRUNK-2	Trunk-2
25	LEAF-1	Leaf-1
26	LEAF-2	Leaf-2
27	RDP	Reliable Data Protocol
28	IRTP	Internet Reliable Transaction
29	ISO-TP4	ISO Transport Protocol Class 4
30	NETBLT	Bulk Data Transfer Protocol
31	MFE-NSP	MFE Network Services Protocol
32	MERIT-INP	MERIT Internodal Protocol
33	SEP	Sequential Exchange Protocol
34	3PC	Third Party Connect Protocol
35	IDPR	Inter-Domain Policy Routing Protocol
36	XTP	XTP
37	DDP	Datagram Delivery Protocol
38	IDPR-CMTP	IDPR Control Message Transport Proto
39	TP++	TP++ Transport Protocol
40	IL	IL Transport Protocol
41	IPv6	Ipv6
42	SDRP	Source Demand Routing Protocol
43	IPv6-Route	Routing Header for IPv6
44	IPv6-Frag	Fragment Header for IPv6
45	IDRP	Inter-Domain Routing Protocol
46	RSVP	Reservation Protocol
47	GRE	General Routing Encapsulation
48	MHRP	Mobile Host Routing Protocol
49	BNA	BNA
50	ESP	Encapsulation Security Payload for IPv6
51	AH	Authentication Header for IPv6
52	I-NLSP	Integrated Net Layer Security TUBA
53	SWIPE	IP with Encryption
54	NARP	NBMA Address Resolution Protocol
55	MOBILE	IP Mobility
56	TLSP	Transport Layer Security Protocol using Kryptonet key management
57	SKIP	SKIP
58	IPv6-ICMP	ICMP for IPv6
59	IPv6-NoNxt	No Next Header for IPv6
60	IPv6-Opts	Destination Options for IPv6
61		Any host internal protocol
62	CFTP	CFTP

Decimal	*Keyword*	*Protocol (continued)*
63		Any local network
64	SAT-EXPAK	SATNET and Backroom EXPAK
65	KRYPTOLAN	Kryptolan
66	RVD	MIT Remote Virtual Disk Protocol
67	IPPC	Internet Pluribus Packet Core
68		Any distributed file system
69	SAT-MON	SATNET Monitoring
70	VISA	VISA Protocol
71	IPCV	Internet Packet Core Utility
72	CPNX	Computer Protocol Network Executive
73	CPHB	Computer Protocol Heart Beat
74	WSN	Wang Span Network
75	PVP	Packet Video Protocol
76	BR-SAT-MON	Backroom SATNET Monitoring
77	SUN-ND	SUN ND PROTOCOL-Temporary
78	WB-MON	WIDEBAND Monitoring
79	WB-EXPAK	WIDEBAND EXPAK
80	ISO-IP	ISO Internet Protocol
81	VMTP	VMTP
82	SECURE-VMTP	SECURE-VMTP
83	VINES	VINES
84	TTP	TTP
85	NSFNET-IGP	NSFNET-IGP
86	DGP	Dissimilar Gateway Protocol
87	TCF	TCF
88	EIGRP	EIGRP
89	OSPFIGP	OSPFIGP
90	Sprite-RPC	Sprite RPC Protocol
91	LARP	Locus Address Resolution Protocol
92	MTP	Multicast Transport Protocol
93	AX.25	AX.25 Frames
94	IPIP	IP-within-IP Encapsulation Protocol
95	MICP	Mobile Internetworking Control Pro.
96	SCC-SP	Semaphore Communications Sec. Pro.
97	ETHERIP	Ethernet-within-IP Encapsulation
98	ENCAP	Encapsulation Header
99		any private encryption scheme
100	GMTP	GMTP
101	IFMP	Ipsilon Flow Management Protocol
102	PNNI	PNNI over IP
103	PIM	Protocol Independent Multicast
104	ARIS	ARIS
105	SCPS	SCPS
106	QNX	QNX

(continues)

Decimal	Keyword	Protocol (continued)
107	A/N	Active Networks
108	IPComp	IP Payload Compression Protocol
109	SNP	Sitara Networks Protocol
110		Compaq-Peer Compaq Peer Protocol
111	IPX-in-IP	IPX in IP
112	VRRP	Virtual Router Redundancy Protocol
113	PGM	PGM Reliable Transport Protocol
114		Any 0-hop protocol
115	L2TP	Layer Two Tunneling Protocol
116	DDX	D-II Data Exchange (DDX)
117	IATP	Interactive Agent Transfer Protocol
118	STP	Schedule Transfer Protocol
119	SRP	SpectraLink Radio Protocol
120	UTI	UTI
121	SMP	Simple Message Protocol
122	SM	SM
123	PTP	Performance Transparency Protocol
124		ISIS over IPv4
125	FIRE	
126	CRTP	Combat Radio Transport Protocol
127	CRUDP	Combat Radio User Datagram
128	SSCOPMCE	
129	IPLT	
130	SPS	Secure Packet Shield
131	PIPE	Private IP Encapsulation within IP
132	SCTP	Stream Control Transmission Protocol
133	FC	Fibre Channel
134–254		Unassigned
255		Reserved

Appendix C

Port Numbers

Port numbers are commonly used to identify an application or destination process and are divided into three ranges: Well Known Ports, Registered Ports, and Dynamic or PrivatePorts. Well Known Ports are those from 0 through 1023. Registered Ports are those from 1024 through 49151. The Dynamic or Private Ports are those from 49152 through 65535. This appendix contains the listing of all registered Well Known port numbers (0 through 1023). These port numbers were originally assigned by the Internet Assigned Numbers Authority (IANA) and are now managed by The Internet Corporation for Assigned Names and Numbers (ICANN). The following table provides a summary of Well Known Port numbers.

Port Assignments:

Keyword	*Decimal*	*Description*
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service Multiplexer
tcpmux	1/udp	TCP Port Service Multiplexer
compressnet	2/tcp	Management Utility
compressnet	2/udp	Management Utility
compressnet	3/tcp	Compression Process
compressnet	3/udp	Compression Process
	4/tcp	Unassigned
	4/udp	Unassigned
rje	5/tcp	Remote Job Entry
rje	5/udp	Remote Job Entry
	6/tcp	Unassigned
	6/udp	Unassigned
echo	7/tcp	Echo
echo	7/udp	Echo

(continues)

Keyword	Decimal	Description (continued)
	8/tcp	Unassigned
	8/udp	Unassigned
discard	9/tcp	Discard
discard	9/udp	Discard
	10/tcp	Unassigned
	10/udp	Unassigned
systat	11/tcp	Active Users
systat	11/udp	Active Users
	12/tcp	Unassigned
	12/udp	Unassigned
daytime	13/tcp	Daytime
daytime	13/udp	Daytime
	14/tcp	Unassigned
	14/udp	Unassigned
	15/tcp	Unassigned [was netstat]
	15/udp	Unassigned
	16/tcp	Unassigned
	16/udp	Unassigned
qotd	17/tcp	Quote of the Day
qotd	17/udp	Quote of the Day
msp	18/tcp	Message Send Protocol
msp	18/udp	Message Send Protocol
chargen	19/tcp	Character Generator
chargen	19/udp	Character Generator
ftp-data	20/tcp	File Transfer [Default Data]
ftp-data	20/udp	File Transfer [Default Data]
ftp	21/tcp	File Transfer [Control]
ftp	21/udp	File Transfer [Control]
ssh	22/tcp	SSH Remote Login Protocol
ssh	22/udp	SSH Remote Login Protocol
telnet	23/tcp	Telnet
telnet	23/udp	Telnet
24/tcp		Any private mail system
	24/udp	Any private mail system
smtp	25/tcp	Simple Mail Transfer
smtp	25/udp	Simple Mail Transfer
	26/tcp	Unassigned
	26/udp	Unassigned
nsw-fe	27/tcp	NSW User System FE
nsw-fe	27/udp	NSW User System FE
	28/tcp	Unassigned
	28/udp	Unassigned

Keyword	Decimal	Description (continued)
msg-icp	29/tcp	MSG ICP
msg-icp	29/udp	MSG ICP
	30/tcp	Unassigned
	30/udp	Unassigned
msg-auth	31/tcp	MSG Authentication
msg-auth	31/udp	MSG Authentication
	32/tcp	Unassigned
	32/udp	Unassigned
dsp	33/tcp	Display Support Protocol
dsp	33/udp	Display Support Protocol
	34/tcp	Unassigned
	34/udp	Unassigned
	35/tcp	Any private printer server
	35/udp	Any private printer server
	36/tcp	Unassigned
	36/udp	Unassigned
time	37/tcp	Time
time	37/udp	Time
rap	38/tcp	Route Access Protocol
rap	38/udp	Route Access Protocol
rlp	39/tcp	Resource Location Protocol
rlp	39/udp	Resource Location Protocol
	40/tcp	Unassigned
	40/udp	Unassigned
graphics	41/tcp	Graphics
graphics	41/udp	Graphics
name	42/tcp	Host Name Server
name	42/udp	Host Name Server
nameserver	42/tcp	Host Name Server
nameserver	42/udp	Host Name Server
nicname	43/tcp	Who Is
nicname	43/udp	Who Is
mpm-flags	44/tcp	MPM FLAGS Protocol
mpm-flags	44/udp	MPM FLAGS Protocol
mpm	45/tcp	Message Processing Module [recv]
mpm	45/udp	Message Processing Module [recv]
mpm-snd	46/tcp	MPM [default send]
mpm-snd	46/udp	MPM [default send]
ni-ftp	47/tcp	NI FTP
ni-ftp	47/udp	NI FTP
auditd	48/tcp	Digital Audit Daemon
auditd	48/udp	Digital Audit Daemon
tacacs	49/tcp	Login Host Protocol (TACACS)

(continues)

Keyword	Decimal	Description (continued)
tacacs	49/udp	Login Host Protocol (TACACS)
re-mail-ck	50/tcp	Remote Mail Checking Protocol
re-mail-ck	50/udp	Remote Mail Checking Protocol
la-maint	51/tcp	IMP Logical Address Maintenance
la-maint	51/udp	IMP Logical Address Maintenance
xns-time	52/tcp	XNS Time Protocol
xns-time	52/udp	XNS Time Protocol
domain	53/tcp	Domain Name Server
domain	53/udp	Domain Name Server
xns-ch	54/tcp	XNS Clearinghouse
xns-ch	54/udp	XNS Clearinghouse
isi-gl	55/tcp	ISI Graphics Language
isi-gl	55/udp	ISI Graphics Language
xns-auth	56/tcp	XNS Authentication
xns-auth	56/udp	XNS Authentication
	57/tcp	Any private terminal access
	57/udp	Any private terminal access
xns-mail	58/tcp	XNS Mail
xns-mail	58/udp	XNS Mail
	59/tcp	Any private file service
	59/udp	Any private file service
	60/tcp	Unassigned
	60/udp	Unassigned
ni-mail	61/tcp	NI MAIL
ni-mail	61/udp	NI MAIL
acas	62/tcp	ACA Services
acas	62/udp	ACA Services
whois++	63/tcp	whois++
whois++	63/udp	whois++
covia	64/tcp	Communications Integrator (CI)
covia	64/udp	Communications Integrator (CI)
tacacs-ds	65/tcp	TACACS-Database Service
tacacs-ds	65/udp	TACACS-Database Service
sql*net	66/tcp	Oracle SQL*NET
sql*net	66/udp	Oracle SQL*NET
bootps	67/tcp	Bootstrap Protocol Server
bootps	67/udp	Bootstrap Protocol Server
bootpc	68/tcp	Bootstrap Protocol Client
bootpc	68/udp	Bootstrap Protocol Client
tftp	69/tcp	Trivial File Transfer
tftp	69/udp	Trivial File Transfer
gopher	70/tcp	Gopher
gopher	70/udp	Gopher

Keyword	Decimal	Description (continued)
netrjs-1	71/tcp	Remote Job Service
netrjs-1	71/udp	Remote Job Service
netrjs-2	72/tcp	Remote Job Service
netrjs-2	72/udp	Remote Job Service
netrjs-3	73/tcp	Remote Job Service
netrjs-3	73/udp	Remote Job Service
netrjs-4	74/tcp	Remote Job Service
netrjs-4	74/udp	Remote Job Service
	75/tcp	Any private dial out service
	75/udp	Any private dial out service
deos	76/tcp	Distributed External Object Store
deos	76/udp	Distributed External Object Store
	77/tcp	Any private RJE service
	77/udp	Any private RJE service
vettcp	78/tcp	vettcp
vettcp	78/udp	vettcp
finger	79/tcp	Finger
finger	79/udp	Finger
http	80/tcp	World Wide Web HTTP
http	80/udp	World Wide Web HTTP
www-http	80/tcp	World Wide Web HTTP
www-http	80/udp	World Wide Web HTTP
hosts2-ns	81/tcp	HOSTS2 Name Server
hosts2-ns	81/udp	HOSTS2 Name Server
xfer	82/tcp	XFER Utility
xfer	82/udp	XFER Utility
mit-ml-dev	83/tcp	MIT ML Device
mit-ml-dev	83/udp	MIT ML Device
ctf	84/tcp	Common Trace Facility
ctf	84/udp	Common Trace Facility
mit-ml-dev	85/tcp	MIT ML Device
mit-ml-dev	85/udp	MIT ML Device
mfcobol	86/tcp	Micro Focus Cobol
mfcobol	86/udp	Micro Focus Cobol
	87/tcp	Any private terminal link
	87/udp	Any private terminal link
kerberos	88/tcp	Kerberos
kerberos	88/udp	Kerberos
su-mit-tg	89/tcp	SU/MIT Telnet Gateway
su-mit-tg	89/udp	SU/MIT Telnet Gateway
dnsix	90/tcp	DNSIX Securit Attribute Token Map
dnsix	90/udp	DNSIX Securit Attribute Token Map
mit-dov	91/tcp	MIT Dover Spooler

(continues)

Keyword	Decimal	Description (continued)
mit-dov	91/udp	MIT Dover Spooler
npp	92/tcp	Network Printing Protocol
npp	92/udp	Network Printing Protocol
dcp	93/tcp	Device Control Protocol
dcp	93/udp	Device Control Protocol
objcall	94/tcp	Tivoli Object Dispatcher
objcall	94/udp	Tivoli Object Dispatcher
supdup	95/tcp	SUPDUP
supdup	95/udp	SUPDUP
dixie	96/tcp	DIXIE Protocol Specification
dixie	96/udp	DIXIE Protocol Specification
swift-rvf	97/tcp	Swift Remote Virtural File Protocol
swift-rvf	97/udp	Swift Remote Virtural File Protocol
tacnews	98/tcp	TAC News
tacnews	98/udp	TAC News
metagram	99/tcp	Metagram Relay
metagram	99/udp	Metagram Relay
newacct	100/tcp	[Unauthorized use]
hostname	101/tcp	NIC Host Name Server
hostname	101/udp	NIC Host Name Server
iso-tsap	102/tcp	ISO-TSAP Class 0
iso-tsap	102/udp	ISO-TSAP Class 0
gppitnp	103/tcp	Genesis Point-to-Point Trans Net
gppitnp	103/udp	Genesis Point-to-Point Trans Net
acr-nema	104/tcp	ACR-NEMA Digital Imag. & Comm. 300
acr-nema	104/udp	ACR-NEMA Digital Imag. & Comm. 300
cso	105/tcp	CCSO name server protocol
cso	105/udp	CCSO name server protocol
csnet-ns	105/tcp	Mailbox Name Nameserver
csnet-ns	105/udp	Mailbox Name Nameserver
3com-tsmux	106/tcp	3COM-TSMUX
3com-tsmux	106/udp	3COM-TSMUX
rtelnet	107/tcp	Remote Telnet Service
rtelnet	107/udp	Remote Telnet Service
snagas	108/tcp	SNA Gateway Access Server
snagas	108/udp	SNA Gateway Access Server
pop2	109/tcp	Post Office Protocol - Version 2
pop2	109/udp	Post Office Protocol - Version 2
pop3	110/tcp	Post Office Protocol - Version 3
pop3	110/udp	Post Office Protocol - Version 3
sunrpc	111/tcp	SUN Remote Procedure Call
sunrpc	111/udp	SUN Remote Procedure Call
mcidas	112/tcp	McIDAS Data Transmission Protocol

Keyword	Decimal	Description (continued)
mcidas	112/udp	McIDAS Data Transmission Protocol
ident	113/tcp	
auth	113/tcp	Authentication Service
auth	113/udp	Authentication Service
audionews	114/tcp	Audio News Multicast
audionews	114/udp	Audio News Multicast
sftp	115/tcp	Simple File Transfer Protocol
sftp	115/udp	Simple File Transfer Protocol
ansanotify	116/tcp	ANSA REX Notify
ansanotify	116/udp	ANSA REX Notify
uucp-path	117/tcp	UUCP Path Service
uucp-path	117/udp	UUCP Path Service
sqlserv	118/tcp	SQL Services
sqlserv	118/udp	SQL Services
nntp	119/tcp	Network News Transfer Protocol
nntp	119/udp	Network News Transfer Protocol
cfdptkt	120/tcp	CFDPTKT
cfdptkt	120/udp	CFDPTKT
erpc	121/tcp	Encore Expedited Remote Pro.Call
erpc	121/udp	Encore Expedited Remote Pro.Call
smakynet	122/tcp	SMAKYNET
smakynet	122/udp	SMAKYNET
ntp	123/tcp	Network Time Protocol
ntp	123/udp	Network Time Protocol
ansatrader	124/tcp	ANSA REX Trader
ansatrader	124/udp	ANSA REX Trader
locus-map	125/tcp	Locus PC-Interface Net Map Ser
locus-map	125/udp	Locus PC-Interface Net Map Ser
nxedit	126/tcp	NXEdit
nxedit	126/udp	NXEdit
unitary	126/tcp	Unisys Unitary Login (prior assignment)
unitary	126/udp	Unisys Unitary Login (prior assignment)
locus-con	127/tcp	Locus PC-Interface Conn Server
locus-con	127/udp	Locus PC-Interface Conn Server
gss-xlicen	128/tcp	GSS X License Verification
gss-xlicen	128/udp	GSS X License Verification
pwdgen	129/tcp	Password Generator Protocol
pwdgen	129/udp	Password Generator Protocol
cisco-fna	130/tcp	cisco FNATIVE
cisco-fna	130/udp	cisco FNATIVE
cisco-tna	131/tcp	cisco TNATIVE
cisco-tna	131/udp	cisco TNATIVE
cisco-sys	132/tcp	cisco SYSMAINT

(continues)

Keyword	Decimal	Description (continued)
cisco-sys	132/udp	cisco SYSMAINT
statsrv	133/tcp	Statistics Service
statsrv	133/udp	Statistics Service
ingres-net	134/tcp	INGRES-NET Service
ingres-net	134/udp	INGRES-NET Service
epmap	135/tcp	DCE endpoint resolution
epmap	135/udp	DCE endpoint resolution
profile	136/tcp	PROFILE Naming System
profile	136/udp	PROFILE Naming System
netbios-ns	137/tcp	NETBIOS Name Service
netbios-ns	137/udp	NETBIOS Name Service
netbios-dgm	138/tcp	NETBIOS Datagram Service
netbios-dgm	138/udp	NETBIOS Datagram Service
netbios-ssn	139/tcp	NETBIOS Session Service
netbios-ssn	139/udp	NETBIOS Session Service
emfis-data	140/tcp	EMFIS Data Service
emfis-data	140/udp	EMFIS Data Service
emfis-cntl	141/tcp	EMFIS Control Service
emfis-cntl	141/udp	EMFIS Control Service
bl-idm	142/tcp	Britton-Lee IDM
bl-idm	142/udp	Britton-Lee IDM
imap	143/tcp	Internet Message Access Protocol
imap	143/udp	Internet Message Access Protocol
uma	144/tcp	Universal Management Architecture
uma	144/udp	Universal Management Architecture
uaac	145/tcp	UAAC Protocol
uaac	145/udp	UAAC Protocol
iso-tp0	146/tcp	ISO-IP0
iso-tp0	146/udp	ISO-IP0
iso-ip	147/tcp	ISO-IP
iso-ip	147/udp	ISO-IP
jargon	148/tcp	Jargon
jargon	148/udp	Jargon
aed-512	149/tcp	AED 512 Emulation Service
aed-512	149/udp	AED 512 Emulation Service
sql-net	150/tcp	SQL-NET
sql-net	150/udp	SQL-NET
hems	151/tcp	HEMS
hems	151/udp	HEMS
bftp	152/tcp	Background File Transfer Program
bftp	152/udp	Background File Transfer Program
sgmp	153/tcp	SGMP
sgmp	153/udp	SGMP

Keyword	Decimal	Description (continued)
netsc-prod	154/tcp	NETSC
netsc-prod	154/udp	NETSC
netsc-dev	155/tcp	NETSC
netsc-dev	155/udp	NETSC
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
knet-cmp	157/tcp	KNET/VM Command/Message Protocol
knet-cmp	157/udp	KNET/VM Command/Message Protocol
pcmail-srv	158/tcp	PCMail Server
pcmail-srv	158/udp	PCMail Server
nss-routing	159/tcp	NSS-Routing
nss-routing	159/udp	NSS-Routing
sgmp-traps	160/tcp	SGMP-TRAPS
sgmp-traps	160/udp	SGMP-TRAPS
snmp	161/tcp	SNMP
snmp	161/udp	SNMP
snmptrap	162/tcp	SNMPTRAP
snmptrap	162/udp	SNMPTRAP
cmip-man	163/tcp	CMIP/TCP Manager
cmip-man	163/udp	CMIP/TCP Manager
cmip-agent	164/tcp	CMIP/TCP Agent
smip-agent	164/udp	CMIP/TCP Agent
xns-courier	165/tcp	Xerox
xns-courier	165/udp	Xerox
s-net	166/tcp	Sirius Systems
s-net	166/udp	Sirius Systems
namp	167/tcp	NAMP
namp	167/udp	NAMP
rsvd	168/tcp	RSVD
rsvd	168/udp	RSVD
send	169/tcp	SEND
send	169/udp	SEND
print-srv	170/tcp	Network PostScript
print-srv	170/udp	Network PostScript
multiplex	171/tcp	Network Innovations Multiplex
multiplex	171/udp	Network Innovations Multiplex
cl/1	172/tcp	Network Innovations CL/1
cl/1	172/udp	Network Innovations CL/1
xyplex-mux	173/tcp	Xyplex
xyplex-mux	173/udp	Xyplex
mailq	174/tcp	MAILQ
mailq	174/udp	MAILQ
vmnet	175/tcp	VMNET

(continues)

Keyword	Decimal	Description (continued)
vmnet	175/udp	VMNET
genrad-mux	176/tcp	GENRAD-MUX
genrad-mux	176/udp	GENRAD-MUX
xdmcp	177/tcp	X Display Manager Control Protocol
xdmcp	177/udp	X Display Manager Control Protocol
nextstep	178/tcp	NextStep Window Server
nextstep	178/udp	NextStep Window Server
bgp	179/tcp	Border Gateway Protocol
bgp	179/udp	Border Gateway Protocol
ris	180/tcp	Intergraph
ris	180/udp	Intergraph
unify	181/tcp	Unify
unify	181/udp	Unify
audit	182/tcp	Unisys Audit SITP
audit	182/udp	Unisys Audit SITP
ocbinder	183/tcp	OCBinder
ocbinder	183/udp	OCBinder
ocserver	184/tcp	OCServer
ocserver	184/udp	OCServer
remote-kis	185/tcp	Remote-KIS
remote-kis	185/udp	Remote-KIS
kis	186/tcp	KIS Protocol
kis	186/udp	KIS Protocol
aci	187/tcp	Application Communication Interface
aci	187/udp	Application Communication Interface
mumps	188/tcp	Plus Five's MUMPS
mumps	188/udp	Plus Five's MUMPS
qft	189/tcp	Queued File Transport
qft	189/udp	Queued File Transport
gacp	190/tcp	Gateway Access Control Protocol
gacp	190/udp	Gateway Access Control Protocol
prospero	191/tcp	Prospero Directory Service
prospero	191/udp	Prospero Directory Service
osu-nms	192/tcp	OSU Network Monitoring System
osu-nms	192/udp	OSU Network Monitoring System
srmp	193/tcp	Spider Remote Monitoring Protocol
srmp	193/udp	Spider Remote Monitoring Protocol
irc	194/tcp	Internet Relay Chat Protocol
irc	194/udp	Internet Relay Chat Protocol
dn6-nlm-aud	195/tcp	DNSIX Network Level Module Audit
dn6-nlm-aud	195/udp	DNSIX Network Level Module Audit
dn6-smm-red	196/tcp	DNSIX Session Mgt Module Audit Redir
dn6-smm-red	196/udp	DNSIX Session Mgt Module Audit Redir

Keyword	Decimal	Description (continued)
dls	197/tcp	Directory Location Service
dls	197/udp	Directory Location Service
dls-mon	198/tcp	Directory Location Service Monitor
dls-mon	198/udp	Directory Location Service Monitor
smux	199/tcp	SMUX
smux	199/udp	SMUX
src	200/tcp	IBM System Resource Controller
src	200/udp	IBM System Resource Controller
at-rtmp	201/tcp	AppleTalk Routing Maintenance
at-rtmp	201/udp	AppleTalk Routing Maintenance
at-nbp	202/tcp	AppleTalk Name Binding
at-nbp	202/udp	AppleTalk Name Binding
at-3	203/tcp	AppleTalk Unused
at-3	203/udp	AppleTalk Unused
at-echo	204/tcp	AppleTalk Echo
at-echo	204/udp	AppleTalk Echo
at-5	205/tcp	AppleTalk Unused
at-5	205/udp	AppleTalk Unused
at-zis	206/tcp	AppleTalk Zone Information
at-zis	206/udp	AppleTalk Zone Information
at-7	207/tcp	AppleTalk Unused
at-7	207/udp	AppleTalk Unused
at-8	208/tcp	AppleTalk Unused
at-8	208/udp	AppleTalk Unused
qmtp	209/tcp	The Quick Mail Transfer Protocol
qmtp	209/udp	The Quick Mail Transfer Protocol
z39.50	210/tcp	ANSI Z39.50
z39.50	210/udp	ANSI Z39.50
914c/g	211/tcp	Texas Instruments 914C/G Terminal
914c/g	211/udp	Texas Instruments 914C/G Terminal
anet	212/tcp	ATEXSSTR
anet	212/udp	ATEXSSTR
ipx	213/tcp	IPX
ipx	213/udp	IPX
vmpwscs	214/tcp	VM PWSCS
vmpwscs	214/udp	VM PWSCS
softpc	215/tcp	Insignia Solutions
softpc	215/udp	Insignia Solutions
CAIlic	216/tcp	Computer Associates Int'l License Server
CAIlic	216/udp	Computer Associates Int'l License Server
dbase	217/tcp	dBASE Unix
dbase	217/udp	dBASE Unix
mpp	218/tcp	Netix Message Posting Protocol

(continues)

Keyword	Decimal	Description (continued)
mpp	218/udp	Netix Message Posting Protocol
uarps	219/tcp	Unisys ARPs
uarps	219/udp	Unisys ARPs
imap3	220/tcp	Interactive Mail Access Protocol v3
imap3	220/udp	Interactive Mail Access Protocol v3
fln-spx	221/tcp	Berkeley rlogind with SPX auth
fln-spx	221/udp	Berkeley rlogind with SPX auth
rsh-spx	222/tcp	Berkeley rshd with SPX auth
rsh-spx	222/udp	Berkeley rshd with SPX auth
cdc	223/tcp	Certificate Distribution Center
cdc	223/udp	Certificate Distribution Center
masqdialer	224/tcp	masqdialer
masqdialer	224/udp	masqdialer
	225–241	Reserved
direct	242/tcp	Direct
direct	242/udp	Direct
sur-meas	243/tcp	Survey Measurement
sur-meas	243/udp	Survey Measurement
inbusiness	244/tcp	inbusiness
inbusiness	244/udp	inbusiness
link	245/tcp	LINK
link	245/udp	LINK
dsp3270	246/tcp	Display Systems Protocol
dsp3270	246/udp	Display Systems Protocol
subntbcst_tftp	247/tcp	SUBNTBCST_TFTP
subntbcst_tftp	247/udp	SUBNTBCST_TFTP
bhfhs	248/tcp	bhfhs
bhfhs	248/udp	bhfhs
	249–255	Reserved
rap	256/tcp	RAP
rap	256/udp	RAP
set	257/tcp	Secure Electronic Transaction
set	257/udp	Secure Electronic Transaction
yak-chat	258/tcp	Yak Winsock Personal Chat
yak-chat	258/udp	Yak Winsock Personal Chat
esro-gen	259/tcp	Efficient Short Remote Operations
esro-gen	259/udp	Efficient Short Remote Operations
openport	260/tcp	Openport
openport	260/udp	Openport
nsiiops	261/tcp	IIOP Name Service over TLS/SSL
nsiiops	261/udp	IIOP Name Service over TLS/SSL
arcisdms	262/tcp	Arcisdms
arcisdms	262/udp	Arcisdms

Keyword	Decimal	Description (continued)
hdap	263/tcp	HDAP
hdap	263/udp	HDAP
bgmp	264/tcp	BGMP
bgmp	264/udp	BGMP
x-bone-ctl	265/tcp	X-Bone CTL
x-bone-ctl	265/udp	X-Bone CTL
sst	266/tcp	SCSI on ST
sst	266/udp	SCSI on ST
td-service	267/tcp	Tobit David Service Layer
td-service	267/udp	Tobit David Service Layer
td-replica	268/tcp	Tobit David Replica
td-replica	268/udp	Tobit David Replica
	269–279	Unassigned
http-mgmt	280/tcp	http-mgmt
http-mgmt	280/udp	http-mgmt
personal-link	281/tcp	Personal Link
personal-link	281/udp	Personal Link
cableport-ax	282/tcp	Cable Port A/X
cableport-ax	282/udp	Cable Port A/X
rescap	283/tcp	rescap
rescap	283/udp	rescap
corerjd	284/tcp	corerjd
corerjd	284/udp	corerjd
	285	Unassigned
fxp-1	286/tcp	FXP-1
fxp-1	286/udp	FXP-1
k-block	287/tcp	K-BLOCK
k-block	287/udp	K-BLOCK
	288–307	Unassigned
novastorbakcup	308/tcp	Novastor Backup
novastorbakcup	308/udp	Novastor Backup
entrusttime	309/tcp	EntrustTime
entrusttime	309/udp	EntrustTime
bhmds	310/tcp	bhmds
bhmds	310/udp	bhmds
asip-webadmin	311/tcp	AppleShare IP WebAdmin
asip-webadmin	311/udp	AppleShare IP WebAdmin
vslmp	312/tcp	VSLMP
vslmp	312/udp	VSLMP
magenta-logic	313/tcp	Magenta Logic
magenta-logic	313/udp	Magenta Logic
opalis-robot	314/tcp	Opalis Robot
opalis-robot	314/udp	Opalis Robot

(continues)

Keyword	Decimal	Description (continued)
dpsi	315/tcp	DPSI
dpsi	315/udp	DPSI
decauth	316/tcp	decAuth
decauth	316/udp	decAuth
zannet	317/tcp	Zannet
zannet	317/udp	Zannet
pkix-timestamp	318/tcp	PKIX TimeStamp
pkix-timestamp	318/udp	PKIX TimeStamp
ptp-event	319/tcp	PTP Event
ptp-event	319/udp	PTP Event
ptp-general	320/tcp	PTP General
ptp-general	320/udp	PTP General
pip	321/tcp	PIP
pip	321/udp	PIP
rtsps	322/tcp	RTSPS
rtsps	322/udp	RTSPS
	323–332	Unassigned
texar	333/tcp	Texar Security Port
texar	333/udp	Texar Security Port
	334–343	Unassigned
pdap	344/tcp	Prospero Data Access Protocol
pdap	344/udp	Prospero Data Access Protocol
pawserv	345/tcp	Perf Analysis Workbench
pawserv	345/udp	Perf Analysis Workbench
zserv	346/tcp	Zebra server
zserv	346/udp	Zebra server
fatserv	347/tcp	Fatmen Server
fatserv	347/udp	Fatmen Server
csi-sgwp	348/tcp	Cabletron Management Protocol
csi-sgwp	348/udp	Cabletron Management Protocol
mftp	349/tcp	mftp
mftp	349/udp	mftp
matip-type-a	350/tcp	MATIP Type A
matip-type-a	350/udp	MATIP Type A
matip-type-b	351/tcp	MATIP Type B
matip-type-b	351/udp	MATIP Type B
bhoetty	351/tcp	bhoetty
bhoetty	351/udp	bhoetty
dtag-ste-sb	352/tcp	DTAG
dtag-ste-sb	352/udp	DTAG
bhoedap4	352/tcp	bhoedap4
bhoedap4	352/udp	bhoedap4
ndsauth	353/tcp	NDSAUTH

Keyword	*Decimal*	*Description (continued)*
ndsauth	353/udp	NDSAUTH
bh611	354/tcp	bh611
bh611	354/udp	bh611
datex-asn	355/tcp	DATEX-ASN
datex-asn	355/udp	DATEX-ASN
cloanto-net-1	356/tcp	Cloanto Net 1
cloanto-net-1	356/udp	Cloanto Net 1
bhevent	357/tcp	bhevent
bhevent	357/udp	bhevent
shrinkwrap	358/tcp	Shrinkwrap
shrinkwrap	358/udp	Shrinkwrap
tenebris_nts	359/tcp	Tenebris Network Trace Service
tenebris_nts	359/udp	Tenebris Network Trace Service
scoi2odialog	360/tcp	scoi2odialog
scoi2odialog	360/udp	scoi2odialog
semantix	361/tcp	Semantix
semantix	361/udp	Semantix
srssend	362/tcp	SRS Send
srssend	362/udp	SRS Send
rsvp_tunnel	363/tcp	RSVP Tunnel
rsvp_tunnel	363/udp	RSVP Tunnel
aurora-cmgr	364/tcp	Aurora CMGR
aurora-cmgr	364/udp	Aurora CMGR
dtk	365/tcp	DTK
dtk	365/udp	DTK
odmr	366/tcp	ODMR
odmr	366/udp	ODMR
mortgageware	367/tcp	MortgageWare
mortgageware	367/udp	MortgageWare
qbikgdp	368/tcp	QbikGDP
qbikgdp	368/udp	QbikGDP
rpc2portmap	369/tcp	rpc2portmap
rpc2portmap	369/udp	rpc2portmap
codaauth2	370/tcp	codaauth2
codaauth2	370/udp	codaauth2
clearcase	371/tcp	Clearcase
clearcase	371/udp	Clearcase
ulistproc	372/tcp	ListProcessor
ulistproc	372/udp	ListProcessor
legent-1	373/tcp	Legent Corporation
legent-1	373/udp	Legent Corporation
legent-2	374/tcp	Legent Corporation
legent-2	374/udp	Legent Corporation

(continues)

Keyword	Decimal	Description (continued)
hassle	375/tcp	Hassle
hassle	375/udp	Hassle
nip	376/tcp	Amiga Envoy Network Inquiry Proto
nip	376/udp	Amiga Envoy Network Inquiry Proto
tnETOS	377/tcp	NEC Corporation
tnETOS	377/udp	NEC Corporation
dsETOS	378/tcp	NEC Corporation
dsETOS	378/udp	NEC Corporation
is99c	379/tcp	TIA/EIA/IS-99 modem client
is99c	379/udp	TIA/EIA/IS-99 modem client
is99s	380/tcp	TIA/EIA/IS-99 modem server
is99s	380/udp	TIA/EIA/IS-99 modem server
hp-collector	381/tcp	hp performance data collector
hp-collector	381/udp	hp performance data collector
hp-managed-node	382/tcp	hp performance data managed node
hp-managed-node	382/udp	hp performance data managed node
hp-alarm-mgr	383/tcp	hp performance data alarm manager
hp-alarm-mgr	383/udp	hp performance data alarm manager
arns	384/tcp	A Remote Network Server System
arns	384/udp	A Remote Network Server System
ibm-app	385/tcp	IBM Application
ibm-app	385/udp	IBM Application
asa	386/tcp	ASA Message Router Object Def.
asa	386/udp	ASA Message Router Object Def.
aurp	387/tcp	Appletalk Update-Based Routing Protocol.
aurp	387/udp	Appletalk Update-Based Routing Protocol
unidata-ldm	388/tcp	Unidata LDM
unidata-ldm	388/udp	Unidata LDM
	389/tcp	Lightweight Directory Access Protocol
ldap	389/udp	Lightweight Directory Access Protocol
uis	390/tcp	UIS
uis	390/udp	UIS
synotics-relay	391/tcp	SynOptics SNMP Relay Port
synotics-relay	391/udp	SynOptics SNMP Relay Port
synotics-broker	392/tcp	SynOptics Port Broker Port
synotics-broker	392/udp	SynOptics Port Broker Port
dis	393/tcp	Data Interpretation System
dis	393/udp	Data Interpretation System
embl-ndt	394/tcp	EMBL Nucleic Data Transfer
embl-ndt	394/udp	EMBL Nucleic Data Transfer
netcp	395/tcp	NETscout Control Protocol
netcp	395/udp	NETscout Control Protocol
netware-ip	396/tcp	Novell Netware over IP

Keyword	Decimal	Description (continued)
netware-ip	396/udp	Novell Netware over IP
mptn	397/tcp	Multi Protocol Trans. Net.
mptn	397/udp	Multi Protocol Trans. Net.
kryptolan	398/tcp	Kryptolan
kryptolan	398/udp	Kryptolan
iso-tsap-c2	399/tcp	ISO Transport Class 2 Non-Control over TCP
iso-tsap-c2	399/udp	ISO Transport Class 2 Non-Control over TCP
work-sol	400/tcp	Workstation Solutions
work-sol	400/udp	Workstation Solutions
ups	401/tcp	Uninterruptible Power Supply
ups	401/udp	Uninterruptible Power Supply
genie	402/tcp	Genie Protocol
genie	402/udp	Genie Protocol
decap	403/tcp	decap
decap	403/udp	decap
nced	404/tcp	nced
nced	404/udp	nced
ncld	405/tcp	ncld
ncld	405/udp	ncld
imsp	406/tcp	Interactive Mail Support Protocol
imsp	406/udp	Interactive Mail Support Protocol
timbuktu	407/tcp	Timbuktu
timbuktu	407/udp	Timbuktu
prm-sm	408/tcp	Prospero Resource Manager Sys. Man.
prm-sm	408/udp	Prospero Resource Manager Sys. Man.
prm-nm	409/tcp	Prospero Resource Manager Node Man.
prm-nm	409/udp	Prospero Resource Manager Node Man.
decladebug	410/tcp	DECLadebug Remote Debug Protocol
decladebug	410/udp	DECLadebug Remote Debug Protocol
rmt	411/tcp	Remote MT Protocol
rmt	411/udp	Remote MT Protocol
synoptics-trap	412/tcp	Trap Convention Port
synoptics-trap	412/udp	Trap Convention Port
smsp	413/tcp	SMSP
smsp	413/udp	SMSP
infoseek	414/tcp	InfoSeek
infoseek	414/udp	InfoSeek
bnet	415/tcp	BNet
bnet	415/udp	BNet
silverplatter	416/tcp	Silverplatter
silverplatter	416/udp	Silverplatter
onmux	417/tcp	Onmux
onmux	417/udp	Onmux

(continues)

Keyword	Decimal	Description (continued)
hyper-g	418/tcp	Hyper-G
hyper-g	418/udp	Hyper-G
ariel1	419/tcp	Ariel
ariel1	419/udp	Ariel
smpte	420/tcp	SMPTE
smpte	420/udp	SMPTE
ariel2	421/tcp	Ariel
ariel2	421/udp	Ariel
ariel3	422/tcp	Ariel
ariel3	422/udp	Ariel
opc-job-start	423/tcp	IBM Operations Planning and Control Start
opc-job-start	423/udp	IBM Operations Planning and Control Start
opc-job-track	424/tcp	IBM Operations Planning and Control Track
opc-job-track	424/udp	IBM Operations Planning and Control Track
icad-el	425/tcp	ICAD
icad-el	425/udp	ICAD
smartsdp	426/tcp	smartsdp
smartsdp	426/udp	smartsdp
svrloc	427/tcp	Server Location
svrloc	427/udp	Server Location
ocs_cmu	428/tcp	OCS_CMU
ocs_cmu	428/udp	OCS_CMU
ocs_amu	429/tcp	OCS_AMU
ocs_amu	429/udp	OCS_AMU
utmpsd	430/tcp	UTMPSD
utmpsd	430/udp	UTMPSD
utmpcd	431/tcp	UTMPCD
utmpcd	431/udp	UTMPCD
iasd	432/tcp	IASD
iasd	432/udp	IASD
nnsp	433/tcp	NNSP
nnsp	433/udp	NNSP
mobileip-agent	434/tcp	MobileIP-Agent
mobileip-agent	434/udp	MobileIP-Agent
mobilip-mn	435/tcp	MobilIP-MN
mobilip-mn	435/udp	MobilIP-MN
dna-cml	436/tcp	DNA-CML
dna-cml	436/udp	DNA-CML
comscm	437/tcp	comscm
comscm	437/udp	comscm
dsfgw	438/tcp	dsfgw
dsfgw	438/udp	dsfgw
dasp	439/tcp	dasp dasp

Keyword	Decimal	Description (continued)
	439/udp	dasp
sgcp	440/tcp	sgcp
sgcp	440/udp	sgcp
decvms-sysmgt	441/tcp	decvms-sysmgt
decvms-sysmgt	441/udp	decvms-sysmgt
cvc_hostd	442/tcp	cvc_hostd
cvc_hostd	442/udp	cvc_hostd
https	443/tcp	http protocol over TLS/SSL
https	443/udp	http protocol over TLS/SSL
snpp	444/tcp	Simple Network Paging Protocol
snpp	444/udp	Simple Network Paging Protocol
microsoft-ds	445/tcp	Microsoft-DS
microsoft-ds	445/udp	Microsoft-DS
ddm-rdb	446/tcp	DDM-RDB
ddm-rdb	446/udp	DDM-RDB
ddm-dfm	447/tcp	DDM-RFM
ddm-dfm	447/udp	DDM-RFM
ddm-ssl	448/tcp	DDM-SSL
ddm-ssl	448/udp	DDM-SSL
as-servermap	449/tcp	AS Server Mapper
as-servermap	449/udp	AS Server Mapper
tserver	450/tcp	TServer
tserver	450/udp	TServer
sfs-smp-net	451/tcp	Cray Network Semaphore server
sfs-smp-net	451/udp	Cray Network Semaphore server
sfs-config	452/tcp	Cray SFS config server
sfs-config	452/udp	Cray SFS config server
creativeserver	453/tcp	CreativeServer
creativeserver	453/udp	CreativeServer
contentserver	454/tcp	ContentServer
contentserver	454/udp	ContentServer
creativepartnr	455/tcp	CreativePartnr
creativepartnr	455/udp	CreativePartnr
macon-tcp	456/tcp	macon-tcp
macon-udp	456/udp	macon-udp
scohelp	457/tcp	scohelp
scohelp	457/udp	scohelp
appleqtc	458/tcp	apple quick time
appleqtc	458/udp	apple quick time
ampr-rcmd	459/tcp	ampr-rcmd
ampr-rcmd	459/udp	ampr-rcmd
skronk	460/tcp	skronk
skronk	460/udp	skronk

(continues)

Keyword	Decimal	Description (continued)
datasurfsrv	461/tcp	DataRampSrv
datasurfsrv	461/udp	DataRampSrv
datasurfsrvsec	462/tcp	DataRampSrvSec
datasurfsrvsec	462/udp	DataRampSrvSec
alpes	463/tcp	alpes
alpes	463/udp	alpes
kpasswd	464/tcp	kpasswd
kpasswd	464/udp	kpasswd
	465	Unassigned
digital-vrc	466/tcp	digital-vrc
digital-vrc	466/udp	digital-vrc
mylex-mapd	467/tcp	mylex-mapd
mylex-mapd	467/udp	mylex-mapd
photuris	468/tcp	proturis
photuris	468/udp	proturis
rcp	469/tcp	Radio Control Protocol
rcp	469/udp	Radio Control Protocol
scx-proxy	470/tcp	scx-proxy
scx-proxy	470/udp	scx-proxy# mondex
	471/tcp	Mondex
mondex	471/udp	Mondex
ljk-login	472/tcp	ljk-login
ljk-login	472/udp	ljk-login
hybrid-pop	473/tcp	hybrid-pop
hybrid-pop	473/udp	hybrid-pop
tn-tl-w1	474/tcp	tn-tl-w1
tn-tl-w2	474/udp	tn-tl-w2
tcpnethaspsrv	475/tcp	tcpnethaspsrv
tcpnethaspsrv	475/udp	tcpnethaspsrv
tn-tl-fd1	476/tcp	tn-tl-fd1
tn-tl-fd1	476/udp	tn-tl-fd1
ss7ns	477/tcp	ss7ns
ss7ns	477/udp	ss7ns
spsc	478/tcp	spsc
spsc	478/udp	spsc
iafserver	479/tcp	iafserver
iafserver	479/udp	iafserver
iafdbase	480/tcp	iafdbase
iafdbase	480/udp	iafdbase
ph	481/tcp	Ph service
ph	481/udp	Ph service
bgs-nsi	482/tcp	bgs-nsi
bgs-nsi	482/udp	bgs-nsi

Keyword	Decimal	Description (continued)
ulpnet	483/tcp	ulpnet
ulpnet	483/udp	ulpnet
integra-sme	484/tcp	Integra Software Management Environment
integra-sme	484/udp	Integra Software Management Environment
powerburst	485/tcp	Air Soft Power Burst
powerburst	485/udp	Air Soft Power Burst
avian	486/tcp	avian
avian	486/udp	avian
saft	487/tcp	saft Simple Asynchronous File Transfer
saft	487/udp	saft Simple Asynchronous File Transfer
gss-http	488/tcp	gss-http
gss-http	488/udp	gss-http
nest-protocol	489/tcp	nest-protocol
nest-protocol	489/udp	nest-protocol
micom-pfs	490/tcp	micom-pfs
micom-pfs	490/udp	micom-pfs
go-login	491/tcp	go-login
go-login	491/udp	go-login
ticf-1	492/tcp	Transport Independent Convergence for FNA
ticf-1	492/udp	Transport Independent Convergence for FNA
ticf-2	493/tcp	Transport Independent Convergence for FNA
ticf-2	493/udp	Transport Independent Convergence for FNA
pov-ray	494/tcp	POV-Ray
pov-ray	494/udp	POV-Ray
intecourier	495/tcp	intecourier
intecourier	495/udp	intecourier
pim-rp-disc	496/tcp	PIM-RP-DISC
pim-rp-disc	496/udp	PIM-RP-DISC
dantz	497/tcp	dantz
dantz	497/udp	dantz
siam	498/tcp	siam
siam	498/udp	siam
iso-ill	499/tcp	ISO ILL Protocol
iso-ill	499/udp	ISO ILL Protocol
isakmp	500/tcp	isakmp
isakmp	500/udp	isakmp
stmf	501/tcp	STMF
stmf	501/udp	STMF
asa-appl-proto	502/tcp	asa-appl-proto
asa-appl-proto	502/udp	asa-appl-proto
intrinsa	503/tcp	Intrinsa
intrinsa	503/udp	Intrinsa
citadel	504/tcp	citadel

(continues)

Keyword	*Decimal*	*Description (continued)*
citadel	504/udp	citadel
mailbox-lm	505/tcp	mailbox-lm
mailbox-lm	505/udp	mailbox-lm
ohimsrv	506/tcp	ohimsrv
ohimsrv	506/udp	ohimsrv
crs	507/tcp	crs
crs	507/udp	crs
xvttp	508/tcp	xvttp
xvttp	508/udp	xvttp
snare	509/tcp	snare
snare	509/udp	snare
fcp	510/tcp	FirstClass Protocol
fcp	510/udp	FirstClass Protocol
passgo	511/tcp	PassGo
passgo	511/udp	PassGo
exec	512/tcp	remote process execution
comsat	512/udp	
biff	512/udp	Used by mail system to notify users of new mail received
login	513/tcp	Remote login via telnet
who	513/udp	Maintains databases showing who's logged in to computer
syslog	514/udp	
printer	515/tcp	spooler
printer	515/udp	spooler
videotex	516/tcp	videotex
videotex	516/udp	videotex
talk	517/tcp	similar to a tenex link
talk	517/udp	
ntalk	518/tcp	
ntalk	518/udp	
utime	519/tcp	unixtime
utime	519/udp	unixtime
efs	520/tcp	extended file name server
router	520/udp	local routing process (on site
ripng	521/tcp	ripng
ripng	521/udp	ripng
ulp	522/tcp	ULP
ulp	522/udp	ULP
ibm-db2	523/tcp	IBM-DB2
ibm-db2	523/udp	IBM-DB2
ncp	524/tcp	NCP
ncp	524/udp	NCP

Keyword	Decimal	Description (continued)
timed	525/tcp	timeserver
timed	525/udp	timeserver
tempo	526/tcp	newdate
tempo	526/udp	newdate
stx	527/tcp	Stock IXChange
stx	527/udp	Stock IXChange
custix	528/tcp	Customer IXChange
custix	528/udp	Customer IXChange
irc-serv	529/tcp	IRC-SERV
irc-serv	529/udp	IRC-SERV
courier	530/tcp	rpc
courier	530/udp	rpc
conference	531/tcp	chat
conference	531/udp	chat
netnews	532/tcp	readnews
netnews	532/udp	readnews
netwall	533/tcp	For emergency broadcasts
netwall	533/udp	For emergency broadcasts
mm-admin	534/tcp	MegaMedia Admin
mm-admin	534/udp	MegaMedia Admin
iiop	535/tcp	iiop
iiop	535/udp	iiop
opalis-rdv	536/tcp	opalis-rdv
opalis-rdv	536/udp	opalis-rdv
nmsp	537/tcp	Networked Media Streaming Protocol
nmsp	537/udp	Networked Media Streaming Protocol
gdomap	538/tcp	gdomap
gdomap	538/udp	gdomap
apertus-ldp	539/tcp	Apertus Technologies Load Determination
apertus-ldp	539/udp	Apertus Technologies Load Determination
uucp	540/tcp	uucpd
uucp	540/udp	uucpd
uucp-rlogin	541/tcp	uucp-rlogin
uucp-rlogin	541/udp	uucp-rlogin
commerce	542/tcp	commerce
commerce	542/udp	commerce
klogin	543/tcp	
klogin	543/udp	
kshell	544/tcp	krcmd
kshell	544/udp	krcmd
appleqtcsrvr	545/tcp	appleqtcsrvr
appleqtcsrvr	545/udp	appleqtcsrvr
dhcpv6-client	546/tcp	DHCPv6 Client

(continues)

Keyword	Decimal	Description (continued)
dhcpv6-client	546/udp	DHCPv6 Client
dhcpv6-server	547/tcp	DHCPv6 Server
dhcpv6-server	547/udp	DHCPv6 Server
afpovertcp	548/tcp	AFP over TCP
afpovertcp	548/udp	AFP over TCP
idfp	549/tcp	IDFP
idfp	549/udp	IDFP
new-rwho	550/tcp	new-who
new-rwho	550/udp	new-who
cybercash	551/tcp	cybercash
cybercash	551/udp	cybercash
deviceshare	552/tcp	deviceshare
deviceshare	552/udp	deviceshare
pirp	553/tcp	pirp
pirp	553/udp	pirp
rtsp	554/tcp	Real Time Stream Control Protocol
rtsp	554/udp	Real Time Stream Control Protocol
dsf	555/tcp	
dsf	555/udp	
remotefs	556/tcp	rfs server
remotefs	556/udp	rfs server
openvms-sysipc	557/tcp	openvms-sysipc
openvms-sysipc	557/udp	openvms-sysipc
sdnskmp	558/tcp	SDNSKMP
sdnskmp	558/udp	SDNSKMP
teedtap	559/tcp	TEEDTAP
teedtap	559/udp	TEEDTAP
rmonitor	560/tcp	rmonitord
rmonitor	560/udp	rmonitord
monitor	561/tcp	
monitor	561/udp	
chshell	562/tcp	chcmd
chshell	562/udp	chcmd
nntps	563/tcp	nntp protocol over TLS/SSL (was snntp)
nntps	563/udp	nntp protocol over TLS/SSL (was snntp)
9pfs	564/tcp	plan 9 file service
9pfs	564/udp	plan 9 file service
whoami	565/tcp	whoami
whoami	565/udp	whoami
streettalk	566/tcp	streettalk
streettalk	566/udp	streettalk
banyan-rpc	567/tcp	banyan-rpc
banyan-rpc	567/udp	banyan-rpc

Keyword	Decimal	Description (continued)
ms-shuttle	568/tcp	microsoft shuttle
ms-shuttle	568/udp	microsoft shuttle
ms-rome	569/tcp	microsoft rome
ms-rome	569/udp	microsoft rome
meter	570/tcp	demon
meter	570/udp	demon
meter	571/tcp	udemon
meter	571/udp	udemon
sonar	572/tcp	sonar
sonar	572/udp	sonar
banyan-vip	573/tcp	banyan-vip
banyan-vip	573/udp	banyan-vip
ftp-agent	574/tcp	FTP Software Agent System
ftp-agent	574/udp	FTP Software Agent System
vemmi	575/tcp	VEMMI
vemmi	575/udp	VEMMI
ipcd	576/tcp	ipcd
ipcd	576/udp	ipcd
vnas	577/tcp	vnas
vnas	577/udp	vnas
ipdd	578/tcp	ipdd
ipdd	578/udp	ipdd
decbsrv	579/tcp	decbsrv
decbsrv	579/udp	decbsrv
sntp-heartbeat	580/tcp	SNTP HEARTBEAT
sntp-heartbeat	580/udp	SNTP HEARTBEAT
bdp	581/tcp	Bundle Discovery Protocol
bdp	581/udp	Bundle Discovery Protocol
scc-security	582/tcp	SCC Security
scc-security	582/udp	SCC Security
philips-vc	583/tcp	Philips Video-Conferencing
philips-vc	583/udp	Philips Video-Conferencing
keyserver	584/tcp	Key Server
keyserver	584/udp	Key Server
imap4-ssl	585/tcp	IMAP4+SSL (use 993 instead)
imap4-ssl	585/udp	IMAP4+SSL (use 993 instead)
password-chg	586/tcp	Password Change
password-chg	586/udp	Password Change
submission	587/tcp	Submission
submission	587/udp	Submission
cal	588/tcp	CAL
cal	588/udp	CAL
eyelink	589/tcp	EyeLink

(continues)

Keyword	Decimal	Description (continued)
eyelink	589/udp	EyeLink
tns-cml	590/tcp	TNS CML
tns-cml	590/udp	TNS CML
http-alt	591/tcp	FileMaker, Inc. - HTTP Alternate (see Port 80)
http-alt	591/udp	FileMaker, Inc. - HTTP Alternate (see Port 80)
eudora-set	592/tcp	Eudora Set
eudora-set	592/udp	Eudora Set
http-rpc-epmap	593/tcp	HTTP RPC Ep Map
http-rpc-epmap	593/udp	HTTP RPC Ep Map
tpip	594/tcp	TPIP
tpip	594/udp	TPIP
cab-protocol	595/tcp	CAB Protocol
cab-protocol	595/udp	CAB Protocol
smsd	596/tcp	SMSD
smsd	596/udp	SMSD
ptcnameservice	597/tcp	PTC Name Service
ptcnameservice	597/udp	PTC Name Service
sco-websrvrmg3	598/tcp	SCO Web Server Manager 3
sco-websrvrmg3	598/udp	SCO Web Server Manager 3
acp	599/tcp	Aeolon Core Protocol
acp	599/udp	Aeolon Core Protocol
ipcserver	600/tcp	Sun IPC server
ipcserver	600/udp	Sun IPC server
	601–605	Unassigned
urm	606/tcp	Cray Unified Resource Manager
urm	606/udp	Cray Unified Resource Manager
nqs	607/tcp	nqs
nqs	607/udp	nqs
sift-uft	608/tcp	Sender-Initiated/Unsolicited File Transfer
sift-uft	608/udp	Sender-Initiated/Unsolicited File Transfer
npmp-trap	609/tcp	npmp-trap
npmp-trap	609/udp	npmp-trap
npmp-local	610/tcp	npmp-local
npmp-local	610/udp	npmp-local
npmp-gui	611/tcp	npmp-gui
npmp-gui	611/udp	npmp-gui
	#	John Barnes <jbarnes@crl.com>
hmmp-ind	612/tcp	HMMP Indication
hmmp-ind	612/udp	HMMP Indication
hmmp-op	613/tcp	HMMP Operation
hmmp-op	613/udp	HMMP Operation
sshell	614/tcp	SSLshell
sshell	614/udp	SSLshell

Keyword	Decimal	Description (continued)
sco-inetmgr	615/tcp	Internet Configuration Manager
sco-inetmgr	615/udp	Internet Configuration Manager
sco-sysmgr	616/tcp	SCO System Administration Server
sco-sysmgr	616/udp	SCO System Administration Server
sco-dtmgr	617/tcp	SCO Desktop Administration Server
sco-dtmgr	617/udp	SCO Desktop Administration Server
dei-icda	618/tcp	DEI-ICDA
dei-icda	618/udp	DEI-ICDA
digital-evm	619/tcp	Digital EVM
digital-evm	619/udp	Digital EVM
sco-websrvrmgr	620/tcp	SCO WebServer Manager
sco-websrvrmgr	620/udp	SCO WebServer Manager
escp-ip	621/tcp	ESCP
escp-ip	621/udp	ESCP
collaborator	622/tcp	Collaborator
collaborator	622/udp	Collaborator
aux_bus_shunt	623/tcp	Aux Bus Shunt
aux_bus_shunt	623/udp	Aux Bus Shunt
cryptoadmin	624/tcp	Crypto Admin
cryptoadmin	624/udp	Crypto Admin
dec_dlm	625/tcp	DEC DLM
dec_dlm	625/udp	DEC DLM
asia	626/tcp	ASIA
asia	626/udp	ASIA
passgo-tivoli	627/tcp	PassGo Tivoli
passgo-tivoli	627/udp	PassGo Tivoli
qmqp	628/tcp	QMQP
qmqp	628/udp	QMQP
3com-amp3	629/tcp	3Com AMP3
3com-amp3	629/udp	3Com AMP3
rda	630/tcp	RDA
rda	630/udp	RDA
ipp	631/tcp	IPP (Internet Printing Protocol)
ipp	631/udp	IPP (Internet Printing Protocol)
bmpp	632/tcp	bmpp
bmpp	632/udp	bmpp
servstat	633/tcp	Service Status update (Sterling Software)
servstat	633/udp	Service Status update (Sterling Software)
ginad	634/tcp	ginad
ginad	634/udp	ginad
rlzdbase	635/tcp	RLZ DBase
rlzdbase	635/udp	RLZ DBase
ldaps	636/tcp	ldap protocol over TLS/SSL (was sldap)

(continues)

Keyword	Decimal	Description (continued)
ldaps	636/udp	ldap protocol over TLS/SSL (was sldap)
lanserver	637/tcp	lanserver
lanserver	637/udp	lanserver
mcns-sec	638/tcp	mcns-sec
mcns-sec	638/udp	mcns-sec
msdp	639/tcp	MSDP
msdp	639/udp	MSDP
entrust-sps	640/tcp	entrust-sps
entrust-sps	640/udp	entrust-sps
repcmd	641/tcp	repcmd
repcmd	641/udp	repcmd
esro-emsdp	642/tcp	ESRO-EMSDP V1.3
esro-emsdp	642/udp	ESRO-EMSDP V1.3
sanity	643/tcp	SANity
sanity	643/udp	SANity
dwr	644/tcp	dwr
dwr	644/udp	dwr
pssc	645/tcp	PSSC
pssc	645/udp	PSSC
ldp	646/tcp	LDP
ldp	646/udp	LDP
dhcp-failover	647/tcp	DHCP Failover
dhcp-failover	647/udp	DHCP Failover
rrp	648/tcp	Registry Registrar Protocol (RRP)
rrp	648/udp	Registry Registrar Protocol (RRP)
aminet	649/tcp	Aminet
aminet	649/udp	Aminet
obex	650/tcp	OBEX
obex	650/udp	OBEX
ieee-mms	651/tcp	IEEE MMS
ieee-mms	651/udp	IEEE MMS
udlr-dtcp	652/tcp	UDLR_DTCP
udlr-dtcp	652/udp	UDLR_DTCP
repscmd	653/tcp	RepCmd
repscmd	653/udp	RepCmd
aodv	654/tcp	AODV
aodv	654/udp	AODV
tinc	655/tcp	TINC
tinc	655/udp	TINC
spmp	656/tcp	SPMP
spmp	656/udp	SPMP
rmc	657/tcp	RMC
rmc	657/udp	RMC

Keyword	Decimal	Description (continued)
tenfold	658/tcp	TenFold
tenfold	658/udp	TenFold
url-rendezvous	659/tcp	URL Rendezvous
url-rendezvous	659/udp	URL Rendezvous
mac-srvr-admin	660/tcp	MacOS Server Admin
mac-srvr-admin	660/udp	MacOS Server Admin
hap	661/tcp	HAP
hap	661/udp	HAP
pftp	662/tcp	PFTP
pftp	662/udp	PFTP
purenoise	663/tcp	PureNoise
purenoise	663/udp	PureNoise
secure-aux-bus	664/tcp	Secure Aux Bus
secure-aux-bus	664/udp	Secure Aux Bus
sun-dr	665/tcp	Sun DR
sun-dr	665/udp	Sun DR
mdqs	666/tcp	
mdqs	666/udp	
doom	666/tcp	doom Id Software
doom	666/udp	doom Id Software
disclose	667/tcp	campaign contribution disclosures - SDR Technologies
disclose	667/udp	campaign contribution disclosures - SDR Technologies
mecomm	668/tcp	MeComm
mecomm	668/udp	MeComm
meregister	669/tcp	MeRegister
meregister	669/udp	MeRegister
vacdsm-sws	670/tcp	VACDSM-SWS
vacdsm-sws	670/udp	VACDSM-SWS
vacdsm-app	671/tcp	VACDSM-APP
vacdsm-app	671/udp	VACDSM-APP
vpps-qua	672/tcp	VPPS-QUA
vpps-qua	672/udp	VPPS-QUA
cimplex	673/tcp	CIMPLEX
cimplex	673/udp	CIMPLEX
acap	674/tcp	ACAP
acap	674/udp	ACAP
dctp	675/tcp	DCTP
dctp	675/udp	DCTP
vpps-via	676/tcp	VPPS Via
vpps-via	676/udp	VPPS Via
vpp	677/tcp	Virtual Presence Protocol

(continues)

Keyword	Decimal	Description (continued)
vpp	677/udp	Virtual Presence Protocol
ggf-ncp	678/tcp	GNU Gereration Foundation NCP
ggf-ncp	678/udp	GNU Generation Foundation NCP
mrm	679/tcp	MRM
mrm	679/udp	MRM
entrust-aaas	680/tcp	entrust-aaas
entrust-aaas	680/udp	entrust-aaas
entrust-aams	681/tcp	entrust-aams
entrust-aams	681/udp	entrust-aams
xfr	682/tcp	XFR
xfr	682/udp	XFR
corba-iiop	683/tcp	CORBA IIOP
corba-iiop	683/udp	CORBA IIOP
corba-iiop-ssl	684/tcp	CORBA IIOP SSL
corba-iiop-ssl	684/udp	CORBA IIOP SSL
mdc-portmapper	685/tcp	MDC Port Mapper
mdc-portmapper	685/udp	MDC Port Mapper
hcp-wismar	686/tcp	Hardware Control Protocol Wismar
hcp-wismar	686/udp	Hardware Control Protocol Wismar
asipregistry	687/tcp	asipregistry
asipregistry	687/udp	asipregistry
realm-rusd	688/tcp	REALM-RUSD
realm-rusd	688/udp	REALM-RUSD
nmap	689/tcp	NMAP
nmap	689/udp	NMAP
vatp	690/tcp	VATP
vatp	690/udp	VATP
msexch-routing	691/tcp	MS Exchange Routing
msexch-routing	691/udp	MS Exchange Routing
hyperwave-isp	692/tcp	Hyperwave-ISP
hyperwave-isp	692/udp	Hyperwave-ISP
connendp	693/tcp	connendp
connendp	693/udp	connendp
ha-cluster	694/tcp	ha-cluster
ha-cluster	694/udp	ha-cluster
ieee-mms-ssl	695/tcp	IEEE-MMS-SSL
ieee-mms-ssl	695/udp	IEEE-MMS-SSL
rushd	696/tcp	RUSHD
rushd	696/udp	RUSHD
uuidgen	697/tcp	UUIDGEN
uuidgen	697/udp	UUIDGEN
	698–703	Unassigned
elcsd	704/tcp	errlog copy/server daemon

Keyword	Decimal	Description (continued)
elcsd	704/udp	errlog copy/server daemon
agentx	705/tcp	AgentX
agentx	705/udp	AgentX
silc	706/tcp	SILC
silc	706/udp	SILC
borland-dsj	707/tcp	Borland DSJ
borland-dsj	707/udp	Borland DSJ
	708	Unassigned
entrust-kmsh	709/tcp	Entrust Key Management Service Handler
entrust-kmsh	709/udp	Entrust Key Management Service Handler
entrust-ash	710/tcp	Entrust Administration Service Handler
entrust-ash	710/udp	Entrust Administration Service Handler
cisco-tdp	711/tcp	Cisco TDP
cisco-tdp	711/udp	Cisco TDP
	712–728	Unassigned
netviewdm1	729/tcp	IBM NetView DM/6000 Server/Client
netviewdm1	729/udp	IBM NetView DM/6000 Server/Client
netviewdm2	730/tcp	IBM NetView DM/6000 send/tcp
netviewdm2	730/udp	IBM NetView DM/6000 send/tcp
netviewdm3	731/tcp	IBM NetView DM/6000 receive/tcp
netviewdm3	731/udp	IBM NetView DM/6000 receive/tcp
	732–740	Unassigned
netgw	741/tcp	netGW
netgw	741/udp	netGW
netrcs	742/tcp	Network based Rev. Cont. Sys.
netrcs	742/udp	Network based Rev. Cont. Sys.
	743	Unassigned
flexlm	744/tcp	Flexible License Manager
flexlm	744/udp	Flexible License Manager
	745–746	Unassigned
fujitsu-dev	747/tcp	Fujitsu Device Control
fujitsu-dev	747/udp	Fujitsu Device Control
ris-cm	748/tcp	Russell Info Sci Calendar Manager
ris-cm	748/udp	Russell Info Sci Calendar Manager
kerberos-adm	749/tcp	kerberos administration
kerberos-adm	749/udp	kerberos administration
rfile	750/tcp	
loadav	750/udp	
kerberos-iv	750/udp	kerberos version iv
pump	751/tcp	
pump	751/udp	
qrh	752/tcp	
qrh	752/udp	

(continues)

Keyword	Decimal	Description (continued)
rrh	753/tcp	
rrh	753/udp	
tell	754/tcp	send
tell	754/udp	send
	755–756	Unassigned
nlogin	758/tcp	
nlogin	758/udp	
con	759/tcp	
con	759/udp	
ns	760/tcp	
ns	760/udp	
rxe	761/tcp	
rxe	761/udp	
quotad	762/tcp	
quotad	762/udp	
cycleserv	763/tcp	
cycleserv	763/udp	
omserv	764/tcp	
omserv	764/udp	
webster	765/tcp	
webster	765/udp	
	766	Unassigned
phonebook	767/tcp	phone
phonebook	767/udp	phone
	768	Unassigned
vid	769/tcp	
vid	769/udp	
cadlock	770/tcp	
cadlock	770/udp	
rtip	771/tcp	
rtip	771/udp	
cycleserv2	772/tcp	
cycleserv2	772/udp	
submit	773/tcp	
notify	773/udp	
rpasswd	774/tcp	
acmaint_dbd	774/udp	
entomb	775/tcp	
acmaint_transd	775/udp	
wpages	776/tcp	
wpages	776/udp	
multiling-http	777/tcp	Multiling HTTP
multiling-http	777/udp	Multiling HTTP

Keyword	Decimal	Description (continued)
	778–779	Unassgined
wpgs	780/tcp	
wpgs	780/udp	
	781–785	Unassigned
concert	786/tcp	Concert
concert	786/udp	Concert
qsc	787/tcp	QSC
qsc	787/udp	QSC
	788–799	Unassigned
mdbs_daemon	800/tcp	
mdbs_daemon	800/udp	
device	801/tcp	
device	801/udp	
	802–809	Unassigned
fcp-udp	810/tcp	FCP
fcp-udp	810/udp	FCP Datagram
	811–827	Unassigned
itm-mcell-s	828/tcp	itm-mcell-s
itm-mcell-s	828/udp	itm-mcell-s
pkix-3-ca-ra	829/tcp	PKIX-3 CA/RA
pkix-3-ca-ra	829/udp	PKIX-3 CA/RA
	830–872	Unassigned
rsync	873/tcp	rsync
rsync	873/udp	rsync
	874–885	Unassigned
iclcnet-locate	886/tcp	ICL coNETion locate server
iclcnet-locate	886/udp	ICL coNETion locate server
iclcnet_svinfo	887/tcp	ICL coNETion server info
iclcnet_svinfo	887/udp	ICL coNETion server info
accessbuilder	888/tcp	AccessBuilder
accessbuilder	888/udp	AccessBuilder
cddbp	888/tcp	CD Database Protocol (unassigned but widespread use)
	889–899	Unassigned
omginitialrefs	900/tcp	OMG Initial Refs
omginitialrefs	900/udp	OMG Initial Refs
smpnameres	901/tcp	SMPNAMERES
smpnameres	901/udp	SMPNAMERES
ideafarm-chat	902/tcp	IDEAFARM-CHAT
ideafarm-chat	902/udp	IDEAFARM-CHAT
ideafarm-catch	903/tcp	IDEAFARM-CATCH
ideafarm-catch	903/udp	IDEAFARM-CATCH
	904–910	Unassigned

(continues)

Keyword	*Decimal*	*Description (continued)*
xact-backup	911/tcp	xact-backup
xact-backup	911/udp	xact-backup
	912–988	Unassigned
ftps-data	989/tcp	ftp protocol, data, over TLS/SSL
ftps-data	989/udp	ftp protocol, data, over TLS/SSL
ftps	990/tcp	ftp protocol, control, over TLS/SSL
ftps	990/udp	ftp protocol, control, over TLS/SSL
nas	991/tcp	Netnews Administration System
nas	991/udp	Netnews Administration System
telnets	992/tcp	telnet protocol over TLS/SSL
telnets	992/udp	telnet protocol over TLS/SSL
imaps	993/tcp	imap4 protocol over TLS/SSL
imaps	993/udp	imap4 protocol over TLS/SSL
ircs	994/tcp	irc protocol over TLS/SSL
ircs	994/udp	irc protocol over TLS/SSL
pop3s	995/tcp	pop3 protocol over TLS/SSL (was spop3)
pop3s	995/udp	pop3 protocol over TLS/SSL (was spop3)
vsinet	996/tcp	vsinet
vsinet	996/udp	vsinet
maitrd	997/tcp	
maitrd	997/udp	
busboy	998/tcp	
puparp	998/udp	
garcon	999/tcp	
applix	999/udp	Applix ac
puprouter	999/tcp	
puprouter	999/udp	
cadlock2	1000/tcp	
cadlock2	1000/udp	
	1001–1009	Unassigned
surf	1010/tcp	surf
surf	1010/udp	surf
1011–1022	Reserved	
	1023/tcp	Reserved
	1023/udp	Reserved

Index

A

ABR, see Adjacent border router
Access-group command, 157
Access lists
  extended, 150
  named, 152, 157
  new capabilities in, 152
  reflexive, 153
  time-based, 155
  types of, 148
ACK bit, 87, 94
Acknowledgment Number fields, 84, 86
Active OPEN function call, 90
Adaptive Differential Pulse Code Modulation (ADPCM), 176
Address(es), see also Internet protocol address(es)
  allocation, 181
  anycast, 180
  architecture, 180
  MAC, 73
  notation, 180
  provider-based, 181
  resolution, 72
    operation, 73
    process, data flow during, 104
  structure, provider-based, 183
  translation process, 121
  types, 180
  unicast, 182
Address Resolution Protocol (ARP), 13, 20, 21, 37, 45, 73
  gratuitous, 75
  Hack, 75
  packet field, 74
  proxy, 75
Adjacent border router (ABR), 138
ADPCM, see Adaptive Differential Pulse Code Modulation
Advanced Research Project Agency (ARPA), 26
Anycast address, 180
Applications, 1–12
  audio and video players, 8–10
  built-in diagnostic tools and, 13
  current, 2
  electronic mail, 2–5
  emerging, 8
  file transfers, 5
  remote terminal access, 5
  virtual private networking, 10–12
  Voice over IP, 10
  Web surfing, 8
Applications and built-in diagnostic tools, 101–119
  diagnostic tools, 107–119
    finger, 118–119
    NSLOOKUP, 114–118
    ping, 107–111
    traceroute, 111–114
  DNS, 101–107
    DNS records, 105–107
    domain name structure, 102–103
    name resolution process, 103–105
    purpose, 101–102
ARP, see Address Resolution Protocol
ARPA, see Advanced Research Project Agency
ASCII text, 30
ATM backbone, 42
Audio
  players, 8
  presentation, transmission of through Internet onto private network, 57
Authentication, 135, 166
Autonomous systems, 122, 125

B

Base Network, 66
Bit positions, decimal values of in byte, 58
Border router (BR), 138
BR, see Border router
Broadcast address, 53, 74
Byte(s)
  decimal values of bit positions in, 58
  octets versus, 39

C

Checksum field, 88, 98
Cisco
  access lists, 158
  Internetwork Operating System (IOS), 152
  router environment, 147
  Systems router, 67
CIXs, see Commercial Internet Exchanges
Code Bits field, 87, 92
Command field, 132
Commercial Internet Exchanges (CIXs), 26
Communications
  Calendar, 8
  protocol, evolution of TCP/IP from, 1
Connection
  establishment, 89
  function calls, 89
Count to infinity, 133
CRC, see Cyclic redundancy check
Cyclic redundancy check (CRC), 17, 45

D

DARPA, see U.S. Department of Defense Advanced Research Projects Agency
Data
  flow
    direction, 148
    during address resolution process, 104
    ports governing, 147
    within TCP/IP network, 23, 24
  streams, 57
  unit, headerless, 19
Datagram, network portion of address in, 68
Decryption, 162
Demilitarized (DMZ) LAN, 49–50, 158
Demultiplexing, 83
Denial-of-service (DoS) attack, 91
Designated router (DR), 138
Destination
  address, 45, 68
  net unreachable message, 113
Diagnostic tools, built in, see Applications and built-in diagnostic tools
Dial-up access, to Remote Access Service server
Differentiated Service (DiffServe), 42
DiffServe, see Differentiated Service
Direct cabling, 142
Distance vector information, 130
DMZ LAN, see Demilitarized LAN
DNS, see Domain Name Service
Domain
  name
    computers having common, 61
    structure, 102
    tree, 102, 103
  suffixes, common, 61
Domain Name Service (DNS), 2, 22, 59, 99, 121
DoS attack, see Denial-of-service attack
Dotted decimal notation, 58
DR, see Designated router
Dynamic ports, 84
Dynamic table updates, 128

E

Echo-reply, 151
EGP, see Exterior Gateway Protocol
Electronic commerce site, 9
Electronic Rolodex, 2
E-mail, 2, 173
Emerging technologies, 14, 163–183
  IPv6, 179–183
    address architecture, 180–183
    overview, 180
  mobile IP, 171–173
    operation, 172–173
    overview, 171–172
  virtual private networking, 163–170
    benefits, 164–166
    limitations, 166–167
    other issues, 167–168
    setting up of remote access service, 168–170
  Voice over IP, 173–179
    constraints, 174–177
    networking configurations, 177–179
Encryption, 167
End-to-end delay, 174
ERP, see Exterior Router Protocol

Error message, inconsistent network mask, 67
Ethernet, 17, 21
  frame format, 72
  LANs, 147
  network, 43
Exterior Gateway Protocol (EGP), 124
Exterior Router Protocol (ERP), 124

F

Federal Bureau of Investigation, 119
FFS, see Firewall Feature Set
File Transfer Protocol (FTP), 5, 22
FIN bit, see Finish bit, 88
Finger
  command, 118
  help screen, 118
Fingering, blocking of, 119
Finish (FIN) bit, 88
Firewall(s), 37, 49
  authentication method, 161
  Feature Set (FFS), 154–154
  functions, 159
  installation, 158
  proxy, 63
  support of proxy services by, 159, 160
Flag field, 43, 44
Fox Network, using RealNetworks RealPlayer to obtains news from, 11
FQDN, see Fully qualified domain name
Fragment Offset fields, 43
Frame Relay, Voice over, 177
FTP, see File Transfer Protocol
Fully qualified domain name (FQDN), 61

G

Gateway, 59, 178
Global timeout value, 154
Gratuitous ARP, 75

H

Hackers, 63, 141
HDLC, see High level Data Link Control
Header
  Checksum field, 45
  Length (Hlen) field, 40, 41, 86
High level Data Link Control (HDLC), 17
Hlen field, see Header Length field
Host address(es), 52
  restriction, 66
  translation of into IP addresses, 101
HTTP, see HyperText Transport Protocol
HyperText Transport Protocol (HTTP), 143, 145

I

IAB, see Internet Activities Board
IANA, see Internet Assigned Numbers Authority
IBM System Network Architecture (SNA), 5
ICANN, see Internet Corporation for Assigned Names and Numbers
ICMP, see Internet Control Message Protocol
IEEE, see Institute of Electrical and Electronic Engineers
IETF, see Internet Engineering Task Force
IGP, see Interior Gateway Protocol
Illustrative network, 128
Inconsistent network mask error message, 67
Information
  advertisement of reachable, 123
  distance vector, 130
Institute of Electrical and Electronic Engineers (IEEE), 17
Interior Gateway Protocol (IGP), 124
Interior Router Protocol (IRP), 124
International Standards Organization (ISO), 13, 15, 17
International Telecommunications Union Telecommunications body (ITU-T), 18
Internet
  connection to via LANs, 52
  evolution, 25
  expansion of, 141
  importance of, 121
  service provider (ISP), 27, 102, 103, 122, 163
  standards track time, 29
  telephony applications, 99
  version numbers, 40
Internet Activities Board (IAB), 27, 50
Internet Assigned Numbers Authority (IANA), 27, 102, 181
Internet Control Message Protocol (ICMP), 20, 21, 38, 76, 107, 187
  Code Field values based on message type, 79–80
  message types, 37
  type and code values, 189–192
  Type Field, 76, 77
  type numbers, 189–192
Internet Corporation for Assigned Names and Numbers (ICANN), 102
Internet Engineering Steering Group, 28
Internet Engineering Task Force (IETF), 27

Internet governing bodies, standards process and, 25–35
Internet governing bodies, 25–28
IAB and IETF, 27
IANA, 27–28
Internet evolution, 25–27
requests for comments, 28–35
accessing RFCs, 29–35
RFC details, 29
standards process, 28–29
Internet protocol (IP), 13, 14, 18, 20, 187, see also Internet protocol and related protocols
datagram, ICMP messages transported via encapsulation within, 76
header, 39
loopback address, with Ping application, 55
mobile, 171
numbers, assigned, 46–48
protocol type field values, 193–196
standardization process, 51
type field values, 193–196
Internet protocol (IP) address(es), 48
classes of, 52
formats, 53
hierarchy, 50, 51
host, 52
mechanism translating host addresses into, 101
multiple interface, 71
predefined, 145
relationship between subnet mask and, 69
reserved, 62
scheme, 50, 65
Internet protocol and related protocols, 37–80
address resolution, 72–76
address resolution operation, 73–75
Ethernet and Token Ring frame formats, 72–73
LAN delivery, 73
proxy ARP, 75
RARP, 75–76
ICMP, 76–80
evolution, 78
ICMP code field, 78
ICMP type field, 76–77
overview, 76
Internet protocol, 38–48
datagrams and datagram transmission, 38
datagrams and segments, 38
IP header, 39–48
routing, 39
IP addressing, 48–72
basic workstation configuration, 58–61
class A addresses, 53–54
class B addresses, 54–56
class C addresses, 56
class D addresses, 56–57
class E addresses, 57
dotted decimal notation, 58
IP addressing scheme, 50–52
multiple interface addresses, 71–72
overview, 49–50
reserved addresses, 62–64
subnetting, 64–70
Internet Society (ISOC), 27
IOS, see Cisco Internetwork Operating System
IP, see Internet Protocol
IPv6, 179
IPX, see Novell NetWare Internetwork Packet Exchange
IRP, see Interior Router Protocol
ISDN port, router, 179
ISO, see International Standards Organization
ISOC, see Internet Society
ISP, see Internet service provider
ITU-T, see International Telecommunications Union Telecommunications body

J

Jitter buffer, 175

L

LAN(s)
address, 23
administrators, 141
connection to Internet via, 52
delivery, 73
demilitarized, 49–50, 158
Ethernet, 21, 147
organizations having multiple, 56
Token Ring, 21
Latency, 174
Layer isolation, 15
Length field, 98
Link
failure, 131
State Advertisement (LSA), 136, 137
LLC, see Logical Link Control
Logical Link Control (LLC), 17
Loopback address, 53r, 62
LSA, see Link State Advertisement

M

MAC, see Media Access Control
Mail server, 106, 116
Maximum transmission unit (MTU), 43
Media Access Control (MAC), 17, 21, 73
Message types, 137, 138
Microsoft
  Outlook, 2, 3, 4
  Ping, 54
  Windows 95, 5
    finger help screen, 118
    FTP application, example of use of, 6
    NT, 108, 109, 168, 169
    operating system, 59
    Telnet application built into, 5
    Tracert, 112
Military Network (MILNET), 26
MILNET, see Military Network
MLD, see Multicast Listener Discovery
Mobile IP, 171
  concept behind, 173
  in operation, 174
  server, 172
MTU, see Maximum transmission unit
Multicast
  group, 57
  Listener Discovery (MLD), 30
Multiple interface addresses, 71
Multiplexing, 83

N

Named access lists, 152, 157
Name resolution process, 103
NAT, see Network address translation
National Science Foundation (NSF), 26
Near-real-time, 10
Netscape Communications, 8
  Communicator, major components of, 9
  Composer, 8
  Navigation, 8
Network(s)
  adapter, 74
  address translation (NAT), 63, 162
  configurations, 177
  Ethernet, 43
  illustrative, 128
  of interconnected networks, 50
  managers, 141
  packet, 175
  prefix, 51
  private, 57, 78
  routing, 122
  security, 64
  service provider (NSP), 103
  Token Ring, 43
  use of VPN via packet, 165
News organization video feeds, 57
Next Hop field, 134
Novell NetWare Internetwork Packet Exchange (IPX), 133
NSF, see National Science Foundation
NSLOOKUP, 114, 115, 116, 117
NSP, see Network service provider

O

Octets, bytes versus, 39
Ohio State University
  Computer and Information Science Department of, 30
  RFC index list, 33
  viewing access to RFCs via computer, 32
Open Shortest Path First (OSPF), 135
  initialization of, 139
  message types, 138
  router(s)
    multicast-enabled, 139
    protocol, 136
    types of, 137
Open Systems Interconnection (OSI) Reference Model, 13, 15, 16
  application layer, 19
  data link layer, 17
  network layer, 17
  physical layer, 16
  presentation layer, 19
  session layer, 18
  transport layer, 18
Options field, 89
OSI Reference Model, see Open Systems Interconnection Reference Model
OSPF, see Open Shortest Path First

P

Packet
  field, RIP version 1, 132
  Internetwork Groper, 107, see also Ping
  network
    operation, 175
    use of VPN via, 165
  subdivision, 177
Padding field, 89
PAR, see Positive Acknowledgment Retransmission

PARC, see Xerox Palo Alto Research Center
Passive OPEN function call, 90
Password, 143
PBX, 178
PCM, see Pulse Code Modulation
PDAs, see Personnel Digital Assistants
Personnel Digital Assistants (PDAs), 50
Ping, 107
  application, IP loopback address with, 55
  command, 108
  implementation, 108
  Microsoft, 54,
  request, 151
Port(s)
  dynamic, 84
  fields, source and destination, 82
  /network table, 126
  numbers, 84, 197–230
  private, 84
  registered, 84
  use, 85
  virtual terminal, 145
  well-known, 84
Positive Acknowledgment Retransmission (PAR), 86
Precedence sub-field, 42
Private ports, 84
Protocol, see also Internet protocol
  field, 44
  stack, 101
Protocol suite, 15–24
  ISO Reference Model, 15–19
    data flow, 19
    layers, 16–19
  TCP/IP protocol suite, 12, 19–24
    application layer, 23
    data flow, 23–24
    network layer, 20–21
    transport layer, 21–23
Provider-based addresses, 181, 183
Proxy
  ARP, 75
  firewall, 63
  services, firewall supported by, 159, 160
PSH bit, see Push bit, 87
PSTN, see Public Switched Telephone Network
Public Switched Telephone Network (PSTN), 176
Pulse Code Modulation (PCM), 176
Push (PSH) bit, 87

Q

QoS, see Quality of Service
Quality of Service (QoS), 42

R

RARP, see Reverse Address Resolution Protocol
RAS server, see Remote Access Service server
RealNetworks RealPlayer, 8, 10, 11
Real Time Protocol (RTP), 175
Reflexive access lists, 153
Registered ports, 84
Remote access service, setting up of, 168
Remote Access Service (RAS) server, 168, 169
  dial-up access to, 170
  TCP/IP configuration, 168, 173
Remote terminal
  access, 5
  transmission, 22
Request for Comments (RFCs), 12, 27, 28
  accessing, 29
  announcements, 30
  categories, 29
  copyright notice, 35
  draft, 28
  index list, Ohio State University, 33
  methods for finding and retrieving, 31
  standard, 29
  viewing access to, 32
Reserved addresses, 62
Reset (RST) bit, 87
Reverse Address Resolution Protocol (RARP), 73, 75
Reverse-network address subnet mask, 149
RFCs, see Request for Comments
RIP, see Routing Information Protocol
Round-trip delay, use of Ping to determine, 110
Router(s), 37, 49
  access
    configurations, 142
    lists, 146
  Cisco Systems, 67
  configuration sub-system, 141
  control, 142
  failure, 114
  interface, assigning multiple network addresses to common, 71

ISDN port, 179
Link State Advertisement, 138
memory cycles, 136
organization's internal, 65
OSPF, multicast-enabled, 139
routing table contents broadcast by, 129
types, 137
using Telnet to access configuration subsystem on, 144
virtual terminal port, 145
voice modules, 177, 178
Route Tag field, 134
Routing, 13, 39
Information Protocol (RIP), 127, 128
protocols, 124, 136
tables, 125, 127
Routing and routing protocol, 121–139
network routing, 122–127
routing in global system, 122–127
routing table update methods, 127
OSPF, 135–139
initialization activity, 136–137
message types, 137–139
operation, 139
overview, 136
path metrics, 136
router types, 137
routing information protocol, 128–135
basic limitations, 131
basic RIPv1 packet, 132–133
dynamic table updates, 128–131
illustrative network, 128
RIPv2, 133–135
RIP versions, 131–132
RST bit, see Reset bit
RTP, see Real Time Protocol

S

Security, 13, 141–162
firewalls, 158–162
basic functions, 159–162
installation location, 158–159
router access considerations, 142–146
direct cabling, 142–143
router control, 142
Telnet and Web access, 143–146
router access lists, 146–158
applying named access list, 157
configuration principles, 158
limitations, 158
new capabilities in access lists, 152–157
rationale for use, 146–148
types of access lists, 148–152
Sender Hardware Address Field, 74
Sequence field, 84
Server
information, protecting, 117
mail, 106
Microsoft Windows NT, 168, 169
mobile IP, 172
Remote Access Service, 168
Whitehouse Web, 113
Yale University, 115, 116
Session termination, 96
Shortest Path First (SPF) protocol, 127
Simple Network Management Protocol (SNMP), 22, 99
Sliding window, 86, 93
Slow start threshold, 95
SNA, see IBM System Network Architecture
SNMP, see Simple Network Management Protocol
SOA, see Start of Authority
Socket, 83
Software developers, third-party, 5
Source
entry, 149
field, 45
SPF protocol, see Shortest Path First protocol
Standards process, see Internet governing bodies, standards process and
Start of Authority (SOA), 106, 116, 117
Sub-domain, 102
Subnet
mask, 59, 60, 68
reference, 70
relationship between IP address and, 69
reverse-network address, 149
viewing, internal versus external, 67
zero, 66
Subnetting
example, 64
illustration of, 65
SYN bit, see Synchronization bit
Synchronization (SYN) bit, 88
SYN-SYN-ACK sequence, 91

T

Target IP Address Field, 74
TCP, see Transmission Control Protocol
TCP/IP, see Transmission Control Protocol/Internet Protocol

Telnet, 22, 90, 143
  application, built into Microsoft Windows, 5
  connection, 82
Three-way handshake, 91, 92, 156
Time-based access lists, 155
Time to live (TTL), 109
  field, 44
  option, 109
  value, 111
Timestamp, 127
Token Ring, 17, 21
  frame format, 72
  networks, 43
ToS field, see Type of Service field
Total Length field, 42
Traceroute, 111
Transmission Control Protocol (TCP), 21, 26, 81
  connection, termination of, 97
  header, 81, 82
  intercept, 156, 157
  originator, 95
  retransmissions, 96
  services, well-known, 85
  slow start, 94
  three-way handshake, 156
  window, 93, 94
Transmission Control Protocol/Internet Protocol (TCP/IP), 19
  configuration, Remote Access Service server, 168, 173
  network(s)
    data flow within, 23, 24
    packet-switched, 38
  Properties dialog box, 60
  protocol suite, 12
    advantage of, 141
    network created using during mid-1980s, 26
    Network Layer Troika of, 37
    role of, 1
  stack, network layer of, 20
Transmission method, obsolete, 38
Transport layer, 81–99
  TCP, 81–96
    checksum field, 88
    code bits field, 87–88
    connection establishment, 89
    connection function cells, 89–91
    Hlen field, 86–87
    options, 89
    padding field, 89
    sequence and acknowledgment number fields, 84–86
    source and destination port fields, 82–84
    TCP header, 81–82
    TCP window, 93–96
    three-way handshake, 91–93
    urgent pointer field, 88–89
  UDP, 96–99
    applications, 99
    operation, 98
    UDP header, 97–98
TTL, see Time to live
Type of Service (ToS) field, 41, 146

U

UDP, see User Datagram Protocol
Unicast addresses, 182
Uniform Resource Locator (URL), 30
UNIX, 108
URG bit, see Urgent bit
Urgent (URG) bit, 87
Urgent Pointer field, 88
URL, see Uniform Resource Locator
U.S. Department of Defense
  Advanced Research Projects Agency (DARPA), 25
  funding by, 25
User Datagram Protocol (UDP), 13, 18, 21, 22, 81
  datagram, 98, 111
  header, 97, 98
  services, well-known, 85
  supported by TCP/IP protocol suite, 96

V

Version field, 132
Video
  feeds, news organization, 57
  presentation, transmission of through Internet onto private network, 57
Virtual private network (VPN), 1, 10, 163
  corporate use of, 166
  creation of, 164
  driving force for adopting, 169
Virtual terminal (vt) port, 145
Voice
  coder, 177
  coding algorithm, 175
  digitized, 99
  gateway, 178
  modules, router, 178
Voice over Frame Relay, 177
Voice over IP (VoIP), 10, 22, 173
VoIP, see Voice over IP

VPN, see Virtual private network
vt port, see Virtual terminal port

W

WAN connections, slow-speed, 94, 105
Web
  browser, 8, 145
  library, 30
  page creation, 8
  surfing, 8, 146
Whitehouse Web server, 113
Window field, 88
Workstation configuration, 58

X

Xerox
  Network Services (XNS), 133
  Palo Alto Research Center (PARC), 72
XNS, see Xerox Network Services

Y

Yale University server, 115

Z

Zero subnet, 66

Network Computing / Telecommunications

TCP/IP Professional Reference Guide

GILBERT HELD

As an IT manager or consultant, you don't need to know the minutiae of every field and every protocol. While it is important to understand how to write code and configure protocols, you hire experts to do just that. You need to understand the issues involved in operating or planning an IP-based network. You need higher levels of information, such as the effect of using DHCP vs. individual network addresses, not the details of configuring DHCP. The ***TCP/IP Professional Reference Guide*** describes the applications and methods, concepts and economics you need to make informed decisions for your organization.

A comprehensive reference, the ***TCP/IP Professional Reference Guide*** provides you with detailed information that covers the concepts associated with the TCP/IP protocol stack, current and emerging applications, security via routers and firewalls, design issues including the use of subnetting, and IPv6. The material covers the various protocols and what they do, how applications operate, addressing issues, security methods, routing, and an overview of emerging technologies.

FEATURES

- Provides detailed information covering the concepts associated with the TCP/IP protocol stack
- Explains current and emerging applications and issues
- Explores issues involved in operating or planning to operate an IP-based network
- Includes network design constraints, testing methods, troubleshooting, and other management issues
- Covers constructing a virtual private network, how to incorporate IP telephony, use voice gateways, and other emerging technologies

AU0824

ISBN 0-8493-0824-0

90000

9 780849 308246